**SSL Certification Installation Guide (CRY)**

SSL Certificate Installation

To check Previous installed SSL cert path

#cd /etc/apache2/sites-available

#cat default-ssl.conf

administrator@Hamel:/etc/apache2/sites-available$ cat default-ssl.conf

<IfModule mod\_ssl.c>

<VirtualHost \_default\_:443>

ServerAdmin webmaster@localhost

DocumentRoot /var/www/html

# Available loglevels: trace8, ..., trace1, debug, info, notice, warn,

# error, crit, alert, emerg.

# It is also possible to configure the loglevel for particular

# modules, e.g.

#LogLevel info ssl:warn

ErrorLog ${APACHE\_LOG\_DIR}/error.log

CustomLog ${APACHE\_LOG\_DIR}/access.log combined

# For most configuration files from conf-available/, which are

# enabled or disabled at a global level, it is possible to

# include a line for only one particular virtual host. For example the

# following line enables the CGI configuration for this host only

# after it has been globally disabled with "a2disconf".

#Include conf-available/serve-cgi-bin.conf

# SSL Engine Switch:

# Enable/Disable SSL for this virtual host.

SSLEngine on

# A self-signed (snakeoil) certificate can be created by installing

# the ssl-cert package. See

# /usr/share/doc/apache2/README.Debian.gz for more info.

# If both key and certificate are stored in the same file, only the

# SSLCertificateFile directive is needed.

SSLCertificateFile /etc/ssl/ssl\_cert/server.cer

SSLCertificateKeyFile /etc/ssl/ssl\_cert/private.key

SSLCertificateChainFile /etc/ssl/ssl\_cert/bundle.cer

# Server Certificate Chain:

# Point SSLCertificateChainFile at a file containing the

# concatenation of PEM encoded CA certificates which form the

# certificate chain for the server certificate. Alternatively

# the referenced file can be the same as SSLCertificateFile

# when the CA certificates are directly appended to the server

# certificate for convinience.

#SSLCertificateChainFile /etc/apache2/ssl.crt/server-ca.crt

# Certificate Authority (CA):

# Set the CA certificate verification path where to find CA

# certificates for client authentication or alternatively one

# huge file containing all of them (file must be PEM encoded)

# Note: Inside SSLCACertificatePath you need hash symlinks

# to point to the certificate files. Use the provided

# Makefile to update the hash symlinks after changes.

#SSLCACertificatePath /etc/ssl/certs/

#SSLCACertificateFile /etc/apache2/ssl.crt/ca-bundle.crt

# Certificate Revocation Lists (CRL):

# Set the CA revocation path where to find CA CRLs for client

# authentication or alternatively one huge file containing all

# of them (file must be PEM encoded)

# Note: Inside SSLCARevocationPath you need hash symlinks

# to point to the certificate files. Use the provided

# Makefile to update the hash symlinks after changes.

#SSLCARevocationPath /etc/apache2/ssl.crl/

#SSLCARevocationFile /etc/apache2/ssl.crl/ca-bundle.crl

# Client Authentication (Type):

# Client certificate verification type and depth. Types are

# none, optional, require and optional\_no\_ca. Depth is a

# number which specifies how deeply to verify the certificate

# issuer chain before deciding the certificate is not valid.

#SSLVerifyClient require

#SSLVerifyDepth 10

# SSL Engine Options:

# Set various options for the SSL engine.

# o FakeBasicAuth:

# Translate the client X.509 into a Basic Authorisation. This means that

# the standard Auth/DBMAuth methods can be used for access control. The

# user name is the `one line' version of the client's X.509 certificate.

# Note that no password is obtained from the user. Every entry in the user

# file needs this password: `xxj31ZMTZzkVA'.

# o ExportCertData:

# This exports two additional environment variables: SSL\_CLIENT\_CERT and

# SSL\_SERVER\_CERT. These contain the PEM-encoded certificates of the

# server (always existing) and the client (only existing when client

# authentication is used). This can be used to import the certificates

# into CGI scripts.

# o StdEnvVars:

# This exports the standard SSL/TLS related `SSL\_\*' environment variables.

# Per default this exportation is switched off for performance reasons,

# because the extraction step is an expensive operation and is usually

# useless for serving static content. So one usually enables the

# exportation for CGI and SSI requests only.

# o OptRenegotiate:

# This enables optimized SSL connection renegotiation handling when SSL

# directives are used in per-directory context.

#SSLOptions +FakeBasicAuth +ExportCertData +StrictRequire

<FilesMatch "\.(cgi|shtml|phtml|php)$">

SSLOptions +StdEnvVars

</FilesMatch>

<Directory /usr/lib/cgi-bin>

SSLOptions +StdEnvVars

</Directory>

# SSL Protocol Adjustments:

# The safe and default but still SSL/TLS standard compliant shutdown

# approach is that mod\_ssl sends the close notify alert but doesn't wait for

# the close notify alert from client. When you need a different shutdown

# approach you can use one of the following variables:

# o ssl-unclean-shutdown:

# This forces an unclean shutdown when the connection is closed, i.e. no

# SSL close notify alert is send or allowed to received. This violates

# the SSL/TLS standard but is needed for some brain-dead browsers. Use

# this when you receive I/O errors because of the standard approach where

# mod\_ssl sends the close notify alert.

# o ssl-accurate-shutdown:

# This forces an accurate shutdown when the connection is closed, i.e. a

# SSL close notify alert is send and mod\_ssl waits for the close notify

# alert of the client. This is 100% SSL/TLS standard compliant, but in

# practice often causes hanging connections with brain-dead browsers. Use

# this only for browsers where you know that their SSL implementation

# works correctly.

# Notice: Most problems of broken clients are also related to the HTTP

# keep-alive facility, so you usually additionally want to disable

# keep-alive for those clients, too. Use variable "nokeepalive" for this.

# Similarly, one has to force some clients to use HTTP/1.0 to workaround

# their broken HTTP/1.1 implementation. Use variables "downgrade-1.0" and

# "force-response-1.0" for this.

# BrowserMatch "MSIE [2-6]" \

# nokeepalive ssl-unclean-shutdown \

# downgrade-1.0 force-response-1.0

</VirtualHost>

</IfModule>

Below is the Path where to paste cert files

#cd /etc/ssl/ssl\_cert

bundle.cer Intermediate.cer private.key root.cer server.cer

#mkdir backup21202021

mv bundle.cer Intermediate.cer private.key root.cer server.cer backup21202021

OR

cp bundle.cer Intermediate.cer private.key root.cer server.cer backup21202021

rm bundle.cer Intermediate.cer private.key root.cer server.cer

Winscp

Copy the files from your local to Ubuntu machine /home/administrator/Downloads Paste it here

#cp file1 fil2 file3 file4 file 5 /etc/ssl/ssl\_cert

#systemctl retstart apache2.service

Centos07 cpanel

/etc/apache2/conf

<IfModule ssl\_module>

SSLEngine on

SSLCertificateFile /home/centos/Certificate\_files/server.cer

SSLCertificateKeyFile /home/centos/Certificate\_files/private.key

SSLCertificateChainFile /home/centos/Certificate\_files/bundle.cer

SSLUseStapling Off

</IfModule>

UseCanonicalName Off

<IfModule security2\_module>

SecRuleEngine On

</IfModule>

<IfModule security3\_module>

modsecurity\_rules 'SecRuleEngine On'

</IfModule>

</VirtualHost>

# # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #

#

# !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

# DO NOT EDIT. AUTOMATICALLY GENERATED. USE INCLUDE FILES IF YOU NEED TO MAKE A CHANGE

# !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

#

# # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #

[root@202-87-36-6 conf]# pwd

/etc/apache2/conf