**Configure Apache TLS security.**

**Configuration Procedure**

Install the **Web Server** package group:

# **yum groupinstall -y "Web server"**

Activate at boot time and start the service:

# **systemctl enable httpd**

# **systemctl start httpd**

Add the **HTTPS** service to the firewall configuration and reload it:

# **firewall-cmd --permanent --add-service=https**

Success

# **firewall-cmd --reload**

Success

Let’s assume your server is called **instructor.example.com**.

Generate a X509 certificate valid for **365** days:

# **openssl req -new -x509 -nodes -out /etc/pki/tls/certs/instructor.example.com.crt -keyout /etc/pki/tls/private/instructor.example.com.key -days 365**

Generating a 2048 bit RSA private key

.....+++

..............+++

writing new private key to '/etc/pki/tls/private/instructor.example.com.key'

-----

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) [XX]:

State or Province Name (full name) []:

Locality Name (eg, city) [Default City]:

Organization Name (eg, company) [Default Company Ltd]:

Organizational Unit Name (eg, section) []:

Common Name (eg, your name or your server's hostname) []:**instructor.example.com**

Email Address []:

Edit the **/etc/httpd/conf.d/ssl.conf** file, search for the **SSLCertificate** string and replace as follows:

**SSLCertificateFile /etc/pki/tls/certs/instructor.example.com.crt**

**SSLCertificateKeyFile /etc/pki/tls/private/instructor.example.com.key**

In the same file, search for the **ServerName** string and replace as follows:

**ServerName instructor.example.com:443**

Check the validity of the **configuration**:

# **httpd -t**

Syntax OK

Or:

# **apachectl configtest**

Syntax OK

Restart the **Apache** webserver:

# **apachectl restart**

Check the virtual host configuration:

# **httpd -D DUMP\_VHOSTS**

VirtualHost configuration:

\*:443                   is a NameVirtualHost

         default server instructor.example.com (/etc/httpd/conf.d/ssl.conf:56)

         port 443 namevhost instructor.example.com (/etc/httpd/conf.d/ssl.conf:56)

         port 443 namevhost instructor.example.com (/etc/httpd/conf.d/ssl.conf:56)

**Optionally**, check the certificate:

# **openssl s\_client -connect localhost:443 -state**

SSL\_connect:before/connect initialization

SSL\_connect:SSLv2/v3 write client hello A

SSL\_connect:SSLv3 read server hello A

depth=0 C = XX, L = Default City, O = Default Company Ltd, CN = instructor.example.com

verify error:num=18:self signed certificate

verify return:1

depth=0 C = XX, L = Default City, O = Default Company Ltd, CN = instructor.example.com

verify return:1

**SSL\_connect:SSLv3 read server certificate A**

**SSL\_connect:SSLv3 read server key exchange A**

**SSL\_connect:SSLv3 read server done A**

**SSL\_connect:SSLv3 write client key exchange A**

**SSL\_connect:SSLv3 write change cipher spec A**

**SSL\_connect:SSLv3 write finished A**

**SSL\_connect:SSLv3 flush data**

**SSL\_connect:SSLv3 read server session ticket A**

**SSL\_connect:SSLv3 read finished A**

**CONNECTED(00000003)**

---

Certificate chain

 0 s:/C=XX/L=Default City/O=Default Company Ltd/CN=instructor.example.com

   i:/C=XX/L=Default City/O=Default Company Ltd/CN=instructor.example.com

---

Server certificate

-----BEGIN CERTIFICATE-----

MIIDkzCCAnugAwIBAgIJAIw+9vpI8jtuMA0GCSqGSIb3DQEBCwUAMGAxCzAJBgNV

BAYTAlhYMRUwEwYDVQQHDAxEZWZhdWx0IENpdHkxHDAaBgNVBAoME0RlZmF1bHQg

Q29tcGFueSBMdGQxHDAaBgNVBAMME2NlbnRvczguZXhhbXBsZS5jb20wHhcNMTQw

ODIwMTQyNDQwWhcNMTUwODIwMTQyNDQwWjBgMQswCQYDVQQGEwJYWDEVMBMGA1UE

BwwMRGVmYXVsdCBDaXR5MRwwGgYDVQQKDBNEZWZhdWx0IENvbXBhbnkgTHRkMRww

GgYDVQQDDBNjZW50b3M4LmV4YW1wbGUuY29tMIIBIjANBgkqhkiG9w0BAQEFAAOC

AQ8AMIIBCgKCAQEA3zu5krRBCOU8+2XBM/dk3fjDqLn439/4lXg9o9LdT4aSAP8e

iJJhM5SoG44nYNYBjVchKCzU6WhpkQ43fMEK3jIFnkxAvldz7zhizA8moI9ewuMj

xnWeVCQMC41Jk4jw2pKitVxt5Lk4SX6bZfvkisHGH/RV6WDaargMrJ8N5Pt80jF0

CnldiKZ8PnqFlqhoHH+aeUvrJXmUzmhCxmjXx4YK6UtZ9pbJIlyzkNnD3XOjHwuC

hnMJNnA3jafD471Lu9nNB5EKSIdwn/scfSuo/fcWlrSpKEE1SEB+qs89R5vPIEmu

IjhXrgIlW6HDo1hSWQDe8/eulChHGRMZJFlMUwIDAQABo1AwTjAdBgNVHQ4EFgQU

+VlrvVt4y6P8G01P0DSW9XwBypUwHwYDVR0jBBgwFoAU+VlrvVt4y6P8G01P0DSW

9XwBypUwDAYDVR0TBAUwAwEB/zANBgkqhkiG9w0BAQsFAAOCAQEAgYYVnrs0GDGj

WHtGfak4Mkhw9DcTp60N8+AQR0mXInSA3oekojnCMqQOlf8HmiVJ6EpNgo+L2mFh

pQzZDTAmrJAODoSAYwavrJcbYwD58LVfAdOmDX2zXemirKFd7mnLQMij8WtRuZ/t

fL5ZpnsIz/iGDSZndFbxqKey6j2sbulsjXHG60INwYF0N5dIhHCo5VeOYz7NEXat

7x2n89eNi2awCdid7ArZDNWAqhLFxRreTN8wTR7t3Y0TN9knm7V4ofPPms3KT0Zk

Op1QIcB80jLx6rkcSq1ghadUUpiRFr5BNlMR0Oul8XWQ4u0B17TKu59wwVNyeizc

vmlt/1L1CQ==

-----END CERTIFICATE-----

subject=/C=XX/L=Default City/O=Default Company Ltd/CN=instructor.example.com

issuer=/C=XX/L=Default City/O=Default Company Ltd/CN=instructor.example.com

---

No client certificate CA names sent

Server Temp Key: ECDH, prime256v1, 256 bits

---

SSL handshake has read 1610 bytes and written 375 bytes

---

New, TLSv1/SSLv3, Cipher is ECDHE-RSA-AES256-GCM-SHA384

Server public key is 2048 bit

Secure Renegotiation IS supported

Compression: NONE

Expansion: NONE

SSL-Session:

    Protocol  : TLSv1.2

    Cipher    : ECDHE-RSA-AES256-GCM-SHA384

    Session-ID: 237566220198BE79A3B0EE9E9D12D3221676329C34F44BF577CC9D77BB6F0C99

    Session-ID-ctx:

    Master-Key: EFA5C1BC2D6C3EBC3928C2339338D31602E7908A70663C9D18AADB683BFC91BD

824D91D857A899A79BF1B95F606FE783

    Key-Arg   : None

    Krb5 Principal: None

    PSK identity: None

    PSK identity hint: None

    TLS session ticket lifetime hint: 300 (seconds)

    TLS session ticket:

    0000 - ef 91 60 0f 59 6f 45 28-0b 1c ac ca f0 ab f7 76   ..`.YoE(.......v

    0010 - c8 fa 8e 79 b6 c8 47 6a-a3 cf 9c 8b 51 43 1c 8c   ...y..Gj....QC..

    0020 - 8b 23 83 0b e1 bc bf 33-65 d2 37 e5 84 15 39 b1   .#.....3e.7...9.

    0030 - 02 a3 4c 0d 65 f7 54 a4-20 1c b1 0a 82 c2 5e 84   ..L.e.T. .....^.

    0040 - 75 92 04 de 3e 09 60 71-6e 20 f9 8e fc 8e af 85   u...>.`qn ......

    0050 - 1d 7f eb 2d 41 ca f0 ff-96 1a 29 e3 ca 9d 7c b6   ...-A.....)...|.

    0060 - 04 84 57 1b ab 78 50 65-c8 ed 0d 7b 6f e3 2d 9c   ..W..xPe...{o.-.

    0070 - 05 d2 73 24 71 89 14 cc-35 59 f5 11 16 80 a3 0d   ..s$q...5Y......

    0080 - 43 b7 53 c3 97 22 25 64-40 eb 42 a0 d3 36 6e 32   C.S.."%d@.B..6n2

    0090 - 2b f6 61 35 76 96 cc 12-76 f3 93 d6 e8 16 54 19   +.a5v...v.....T.

    00a0 - 7d 9d a2 50 b1 d5 87 12-61 f7 d4 c1 46 19 23 f5   }..P....a...F.#.

    00b0 - 41 71 43 32 89 7f 9c 9f-b6 ab e3 71 14 d6 13 f4   AqC2.......q....

    Start Time: 1408555281

    Timeout   : 300 (sec)

    Verify return code: 18 (self signed certificate)

---

read:errno=0

SSL3 alert write:warning:close notify

Note: According to **Sander van Vugt**, the **elinks** command doesn’t work well with **TLS** and shouldn’t be used in this specific context.