Linux - Setting up an SSL secured Webserver with CentOS

> Last verified: October 2026 for RHEL / Rocky Linux 9 against the Red Hat Apache TLS guide, the Apache 2.4 upgrade notes and the OpenSSL req manual.

This guide will explain how to set up a site over https. The tutorial uses a self signed key so will work well for a personal website or testing purposes. This is provided as is so proceed at your own risk and take backups!

1. Getting the required software

For an SSL encrypted web server you will need a few things. Depending on your install you may or may not have OpenSSL and mod_ssl, Apache's interface to OpenSSL. Use dnf to get them if you need them (RHEL / Rocky Linux 9 and later; on CentOS 7 the equivalent was yum).

```
dnf install mod_ssl openssl
```

dnf will either tell you they are installed or will install them for you.

2. Generate a self-signed certificate

Using OpenSSL we will generate a self-signed certificate. If you are using this on a production server you are probably likely to want a key from a Trusted Certificate Authority, but if you are just using this on a personal site or for testing purposes a self-signed certificate is fine. To create the key you will need to be root so you can either su to root or use sudo in front of the commands

```
# Generate a private key and a self-signed certificate (valid 365 days) in one step.
# Replace yoursite.com with your server name; browsers check the subjectAltName.
openssl req -x509 -newkey rsa:2048 -noenc -days 365 -keyout ca.key -out ca.crt -subj "/CN=yoursite.com" -
addext "subjectAltName=DNS:yoursite.com"

# Copy the files to the correct locations and protect the key
cp ca.crt /etc/pki/tls/certs/ca.crt
cp ca.key /etc/pki/tls/private/ca.key
chmod 600 /etc/pki/tls/private/ca.key
```

	WARNING: Make sure that you **copy** the files and do not **move** them if you use SELinux. Apache will complain about missing certificate files otherwise, as it cannot read them because the certificate files do not have the right SELinux context.

If you have moved the files and not copied them, you can use the following command to correct the SELinux contexts on those files, as the correct context definitions for /etc/pki/* come with the bundled SELinux policy.

```
restorecon -RvF /etc/pki
```

Then we need to update the Apache SSL configuration file

```
vi +/SSLCertificateFile /etc/httpd/conf.d/ssl.conf
```

Change the paths to match where the Key file is stored. If you've used the method above it will be

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

Then set the correct path for the Certificate Key File a few lines below. If you've followed the instructions above it is:

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

Quit and save the file and then restart Apache

```
systemctl restart httpd
```

All being well you should now be able to connect over https to your server and see the default Apache test page. As the certificate is self signed browsers will generally ask you whether you want to accept the certificate.

3. Setting up the virtual hosts

Just as you set VirtualHosts for http on port 80 so you do for https on port 443. Put them in a file under /etc/httpd/conf.d/ (for example yoursite.conf). A typical VirtualHost for a site on port 80 looks like this

```
<VirtualHost *:80>
        <Directory /var/www/vhosts/yoursite.com/httpdocs>
        AllowOverride All
        </Directory>
        DocumentRoot /var/www/vhosts/yoursite.com/httpdocs
        ServerName yoursite.com
</VirtualHost>
```

To add a sister site on port 443, add a VirtualHost record something like this. Apache 2.4 no longer needs a NameVirtualHost line (the directive has no effect), so do not add one.

```
<VirtualHost *:443>
        SSLEngine on
        SSLCertificateFile /etc/pki/tls/certs/ca.crt
        SSLCertificateKeyFile /etc/pki/tls/private/ca.key
        <Directory /var/www/vhosts/yoursite.com/httpsdocs>
        AllowOverride All
        </Directory>
        DocumentRoot /var/www/vhosts/yoursite.com/httpsdocs
        ServerName yoursite.com
</VirtualHost>
```

Restart Apache again using

```
systemctl restart httpd
```

4. Configuring the firewall

You should now have a site working over https using a self-signed certificate. If you can't connect you may need to open the port on your firewall. RHEL / Rocky Linux 9 use firewalld:

```
firewall-cmd --permanent --add-port=443/tcp
firewall-cmd --reload
firewall-cmd --list-ports
```

Sources

- https://docs.redhat.com/en/documentation/red_hat_enterprise_linux/9/html/deploying_web_servers_and_reverse_proxies/setting-apache-http-server_deploying-web-servers-and-reverse-proxies
- https://httpd.apache.org/docs/2.4/upgrading.html
- https://httpd.apache.org/docs/2.4/vhosts/
- https://docs.openssl.org/3.0/man1/openssl-req/

Revision #1
Created 2 September 2024 20:15:49 by Admin
Updated 24 October 2026 14:10:46 by Admin