How To Configure SSL/TLS With Nagios Fusion 2026R2

Purpose

This document describes how to configure SSL/TLS with Nagios Fusion 2026R2 for secure access to the web user interface.

Note that there are separate steps if your objective is to set up TLS for the Event-Based collection method. You can find full details on that process here:

Configuring TLS for Event-Based Collection in Nagios Fusion

Terminology

For your information:

- SSL = Secure Sockets Layer
- TLS = Transport Layer Security

TLS replaces SSL, however, the tools used to implement both generally use SSL in their name/directives. For simplicity reasons, the rest of this document will use the term SSL.

To implement SSL, you need to generate a certificate. When you generate a certificate, you create a request that needs to be signed by a Certificate Authority (CA). This CA can be:

- A trusted company like VeriSign
- An internal CA that is part of your IT infrastructure, like Microsoft Windows CA
- The Nagios Fusion server itself (self-signed)

The CA will then provide you with a signed certificate. This documentation can used to generate a request that can be submitted to any of these CA types.

Editing Files

In many steps of this documentation, you will be required to edit files. This documentation will use the vi text editor. When using the vi editor:

- To make changes press `i` on the keyboard first to enter insert mode
- Press **Esc** to exit insert mode
- When you have finished, save the changes in vi by typing `:wq` and press **Enter**

Installing Necessary Components

Establish a terminal session with your Nagios Fusion server and as **root** and execute the following command:

RHEL | CentOS | Oracle Linux

```
yum install -y mod_ssl openssl
```

Debian | Ubuntu

```
apt install -y openssl
```

Certificate Directory

The steps in this documentation will be performed from within the `/etc/fusion/certs` directory as **root**. Execute the following command to move to the directory:

```
cd /etc/fusion/certs
```

You will continue to use this terminal session throughout this documentation.

Generate Private Key File

The first step is to generate the private key file, execute the following command:

```
openssl genrsa -out fusion.key 2048
```

That would have generated some random text.

Generate Certificate Request File

Next, generate the certificate request file by executing the following command:

```
openssl req -new -key fusion.key -out fusion.csr -addext "subjectAltName = DNS:[your_fqdn_here]"
```

You will need to supply some values, some can be left blank, however the most important values will be the **Common Name** and the `subjectAltName` or **Subject Alternative Name (SAN)**.

In the example below, `nf-c7x-x64.domain.local` is used as the hostname. When accessing Nagios Fusion in your web browser, this is the hostname you should use. Modern browsers validate the hostname against the certificate Subject Alternative Name (SAN) entry, and if they do not match, certificate warnings may be displayed.

Depending on your environment, the Common Name (CN) and SAN entries may differ, however using the same hostname for both is generally recommended.

More detailed information about certificate hostname validation can be found in the following article:

SSL/TLS - Understanding Certificate Warnings

Example

```
Country Name (2 letter code) [XX]:AU
State or Province Name (full name) []:NSW
Locality Name (city) [Default City]:Sydney
Organization Name (company) [Default Company Ltd]:My Company Pty Ltd
Organizational Unit Name (section) []:
Common Name (your name or your server's hostname) []:nf-c7x-x64.domain.local
Email Address []:
Please enter the following 'extra' attributes to be sent with your certificate request
A challenge password []:
An optional company name []:
```

In the example above, the **Organizational Unit Name** (OU), **Email Address**, **challenge password**, and **optional company name** fields were left blank because they are optional. In particular, a challenge password is not required and can be omitted.

Sign Certificate Request

At this point you have created a certificate request that needs to be signed by a CA.

Using A Trusted CA Company

If you are going to use a trusted company like VeriSign to provide you with a certificate you will need to send them a copy of the certificate request. This can be viewed by executing the following command:

```
cat fusion.csr
```

You'll get a lot of random text; this is what you will need to provide to your trusted CA. You must provide the CA with everything including the `-----BEGIN CERTIFICATE REQUEST-----` and `-----END CERTIFICATE REQUEST-----` lines.

Once they send you the signed certificate you will need to copy the certificate into a new file called `fusion.crt`. The certificate you receive will also be a lot of random text, so you can just paste that text into the new file which you can open with the vi editor:

```
vi fusion.crt
```

You must paste everything including the `-----BEGIN CERTIFICATE -----` and `-----END CERTIFICATE -----` lines when pasting them into the file.

Save the file and close vi.

You can now proceed to the Set Permissions section of this document.

Using Microsoft Windows CA

If you are going to use a Microsoft Windows CA to sign your certificate request, please follow the steps in this Doc Hub article:

SSL/TLS - Signing Certificates With A Microsoft Certificate Authority

After following the KB article, you will have the `fusion.crt` file and you can proceed to the Set Permissions section of this document.

Self-Signing The Certificate

You can also self-sign the certificate by executing the following command:

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

This should produce output saying the Signature was OK and it was Getting Private Key.

Note: When you self-sign a certificate you will get warnings in your web browser.

More detailed information about this can be found in the following Doc Hub article:

SSL/TLS - Understanding Certificate Warnings

Set Permissions

You need to set the correct permissions on the certificate file. Execute the following commands:

```
chmod go-rwx fusion.*
```

Run the setup-ssl.sh Script

The `setup-ssl.sh` script will handle the following tasks for you:

- Updating the Apache Configuration
- Enabling SSL
- Restarting the Apache service

To execute the script, run the following commands:

```
cd /usr/local/fusion/scripts
./setup-ssl.sh
```

Firewall Rules

The following firewall rules **may** need to be added. If you cannot access the Nagios Fusion in the next step (Test Certificate) then you will likely need to run these commands. You can check which ports are open with a command like the following on RHEL, CentOS, and Oracle linux:

```
firewall-cmd --list-ports
```

If you see `443/tcp` in the output as seen in this example, no further steps are necessary:

```
80/tcp 443/tcp 8080/tcp
```

If you do **not** see `443`, you can run the following commands to open port 443:

RHEL | CentOS | Oracle Linux

```
firewall-cmd --zone=public --add-port=443/tcp
firewall-cmd --zone=public --add-port=443/tcp --permanent
```

Debian

The local firewall is not enabled on Debian by default and no steps are required here. **IF** it is enabled then the commands are:

```
iptables -I INPUT -p tcp --destination-port 443 -j ACCEPT
```

Ubuntu

The local firewall is not enabled on Ubuntu by default and no steps are required here. **IF** it is enabled then the commands are:

```
sudo ufw allow https
sudo ufw reload
```

Test Certificate

Now test your connection to the server by directing your web browser to:

```
https://your_server_address/
```

Note: There is no `nagiosfusion/` extension in the URL, we are just testing a connection to Apache to see if the certificate works.

You may get a self-signed certificate warning, but that is OK, you can just add a security exception. If it works, you'll see the Nagios Fusion welcome page.

If it returns an error, check your firewall and backtrack through this document, making sure you've performed all the steps listed.

Notes On Redirecting

With this configuration, if a user types `http://nagiosfusion` in their web browser, it will redirect them to `https://nagiosfusion` which can cause certificate warnings in certain scenarios. If you wanted to redirect them to `https://nagiosfusion.yourdomain.com`, then you simply need to change the `RewriteRule` in the `/etc/httpd/conf/httpd.conf` file:

```
RewriteRule (.*) https://nagiosfusion.yourdomain.com%{REQUEST_URI}
```

Then restart the `httpd` service.

More detailed information about this can be found in the following Doc Hub article:

SSL/TLS - Understanding Certificate Warnings

Finishing Up

This completes the documentationon Configuring SSL/TLS with Nagios Fusion 2026R2. If you have additional questions or other support-related questions, please visit the Nagios Support Forum, Nagios Documentation Hub, or Nagios Library:

Visit Nagios Support Forum Visit Documentation Hub Visit Nagios Library