

# Configuring SSL with Nagios Log Server 2024

## 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 a Microsoft Windows CA
- The Nagios Log Server itself (self signed)

The CA will then provide you with a signed certificate.

This documentation can be 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 to your Nagios Log Server and as root and execute the following command:

**RHEL | CentOS | Oracle Linux**

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

**Debian | Ubuntu**

```
apt-get install -y openssl
```

## Certificate Directory

The steps in this documentation will be performed from within the `/usr/local/nagioslogserver/var/certs/` directory. Execute the following commands to create the directory (if it doesn't exist) and then change into it:

```
mkdir -p /usr/local/nagioslogserver/var/certs
```

```
cd /usr/local/nagioslogserver/var/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 nagioslogserver.key 2048
```

That would have generated some random text.

## Generate Certificate Request File

Next you will generate the certificate request file by executing the following command:

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

You will need to supply some values, some can be left blank, however the most important value is the Common Name. In the example below you can see that nls-c7x-x64.domain.local has been used which means that when you access the Nagios Log Server in your web browser, this is the address you will need to use. This is particularly important, if these don't match then you will get warnings in your web browser. More detailed information about this can be found in the following

KB [article](#).

The following is an example:

```
Country Name (2 letter code) [XX]:AU
```

```
State or Province Name (full name) []:NSW
```

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

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

```
Organizational Unit Name (eg, section) []:
```

```
Common Name (eg, your name or your server's hostname) []:nls-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 []:
```

As you can see above, I did not supply an Organizational Unit Name, email address, password or optional company name. Specifically, providing a password is not necessary.

## 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 nagioslogserver.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 `nagioslogserver.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 nagioslogserver.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 A Microsoft Windows CA

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

[KB article](#).

After following the KB article you will have the `nagioslogserver.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 nagioslogserver.csr -signkey nagios-  
logserver.key -out nagioslogserver.crt
```

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



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 [KB article](#).

## Set Permissions

You need to set permissions on the files, execute the following commands:

```
chmod go-rwx nagioslogserver.*
```

## Update Apache Configuration

Now you have to tell the Apache web server about the certificate. The configuration file for this differs depending on your operating system (OS), open the SSL file in vi by executing the following command:

### RHEL | CentOS | Oracle Linux

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

### Debian | Ubuntu

```
vi /etc/apache2/sites-available/default-ssl.conf
```

Find these lines and update them as follows:

SSLCertificateFile /usr/local/nagioslogserver/var/certs/nagioslogserver.crt

SSLCertificateKeyFile /usr/local/nagioslogserver/var/certs/nagioslogserver.key



Typing /eFile and pressing Enter in vi should take you directly to this section in the file.

In that same file, navigate to the end (press SHIFT + G) and before the line `</VirtualHost>` add the following lines:

```
<IfModule mod_rewrite.c>
```

```
RewriteEngine on
```

```
RewriteCond $1 !^(index\.php|scripts|-  
media|app|js|css|img|font|vendor|config.js)
```

```
RewriteCond %{REQUEST_FILENAME} !-f
```

```
RewriteCond %{REQUEST_FILENAME} !-d
```

```
RewriteRule nagioslogserver/(.*)$ /var/www/htm-  
l/nagioslogserver/www/index.php/$1 [L,QSA]
```

```
</IfModule>
```

Save the changes, you have finished editing this file.

You have to update Apache web server config file to force SSL to be used. The configuration file for this differs depending on your OS, open the SSL file in vi by executing the following command:

**RHEL | CentOS | Oracle Linux**

```
vi /etc/httpd/conf.d/nagioslogserver.conf
```

## Debian | Ubuntu

```
vi /etc/apache2/sites-available/nagioslogserver.conf
```

Add the following lines to the end of the file (press SHIFT + G):

```
RewriteEngine on
```

```
RewriteCond $1 !^(index\.php|scripts|-  
media|app|js|css|img|font|vendor|config.js)
```

```
RewriteCond %{REQUEST_FILENAME} !-f
```

```
RewriteCond %{REQUEST_FILENAME} !-d
```

```
RewriteRule nagioslogserver/(.*)$ /var/www/htm-  
l/nagioslogserver/www/index.php/$1 [L,QSA]
```

```
RewriteCond %{HTTPS} off
```

```
RewriteRule (.*?) https://%{HTTP_HOST}%{REQUEST_URI}
```

It is most likely that you only need to add the two lines in bold above, the end result is that all the lines need to exist.

Save the changes, you have finished editing this file.



## Restart Apache

You need to restart the Apache for the new certificate key to be used.

### RHEL | CentOS | Oracle Linux

```
systemctl restart httpd.service
```

### Debian | Ubuntu

```
a2ensite default-ssl
```

```
a2enmod ssl
```

```
systemctl restart apache2.service
```

## Firewall Rules

The following firewall rules may need to be added. If you cannot access the Nagios Log Server in the next step (Test Certificate) then it's likely you'll need to run these commands:

### 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://yourservername/>



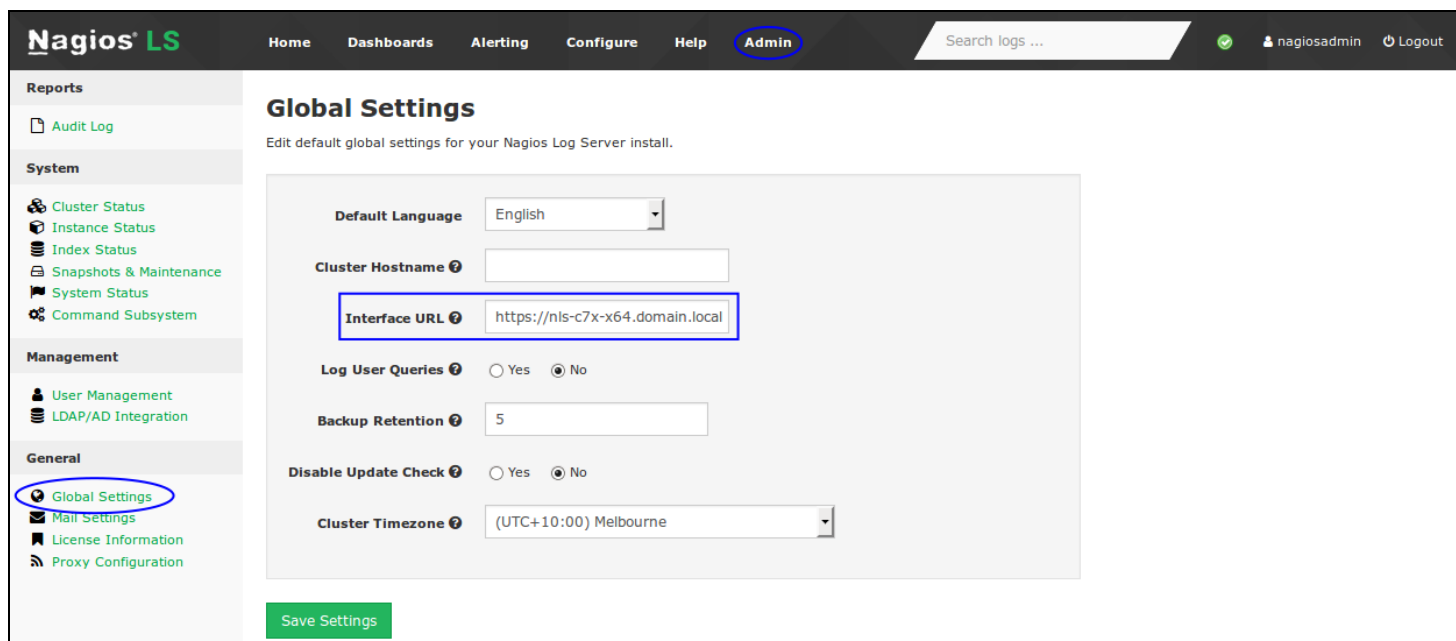
There is no nagioslogserver/ extension in the URL, we are just testing a connection to Apache to see if the certificate works.

You may get a self signed certificate warning, but that is OK, you can just add a security exception. If is working you'll see the Nagios Log Server welcome page. More detailed information about this can be found in the following

[KB article](#). If it returns an error check your firewall and backtrack through this document, making sure you've performed all the steps listed.

## Update Nagios Log Server Configuration

The Nagios Log Server GUI settings also need updating. Open up the Nagios Log Server interface to <https://yourservername/nagioslogserver/> and navigate to Admin > General > Global Settings.



The screenshot shows the Nagios Log Server web interface. The top navigation bar includes links for Home, Dashboards, Alerting, Configure, Help, and Admin (which is circled in blue). A search bar and user information (nagiosadmin) are on the right. The left sidebar has a 'General' section with 'Global Settings' circled in blue. The main content area is titled 'Global Settings' and contains several configuration fields: 'Default Language' (English), 'Cluster Hostname' (empty), 'Interface URL' (https://nls-c7x-x64.domain.local, highlighted with a blue box), 'Log User Queries' (No), 'Backup Retention' (5), 'Disable Update Check' (No), and 'Cluster Timezone' ((UTC+10:00) Melbourne). A 'Save Settings' button is at the bottom.

Change the Interface URL to https instead of the default http and click the Save Settings button.



It's very important that the IP Address / DNS name is the same here as it was typed in the certificate key "common name".

You are now set to use https with your Nagios Log Server web interface.

## Notes On Redirecting

With this configuration, if a user types <http://logserver> in their web browser, it will redirect them to <https://logserver> which can cause certificate warnings in certain scenarios. If you wanted to redirect them to <https://logserver.yourdomain.com> then you simply need to change

the RewriteRule in the /etc/httpd/conf/httpd.conf file:

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

Then restart the httpd service.

More detailed information about this can be found in the following

[KB article](#).

## Repeat On All Instances

This procedure needs to be applied to all instances in your Nagios Log Server cluster to ensure you always have an encrypted connection. You can copy the files from this server to your other instances and then follow the [Update Apache Configuration](#), [Restart Apache](#), and [Firewall Rules](#) sections.



You should create the certificate request with a generic Common Name like nls.-domain.local and create multiple DNS A records for nls.domain.local that are the IP addresses for each server instance.