Configuring SSL

Created by Dulani Wallace, last modified by Joe Tom on Oct 31, 2017

Using Secure Socket Layer (SSL) communication with the repository manager is an important security feature and a recommended best practice. Secure communication can be inbound or outbound.

Outbound client communication may include integration with:

- a remote proxy repository over HTTPS - documented in Repository Management
- SSL/TLS secured servers - e.g. for SMTP/email integration documented in Email Server
- LDAP servers configured to use LDAPS
- specialized authentication realms such as the Crowd realm.

Inbound client communication includes

- web browser HTTPS access to the user interface,
- tool access to repository content,
- and manual or scripted usage of the REST APIs.

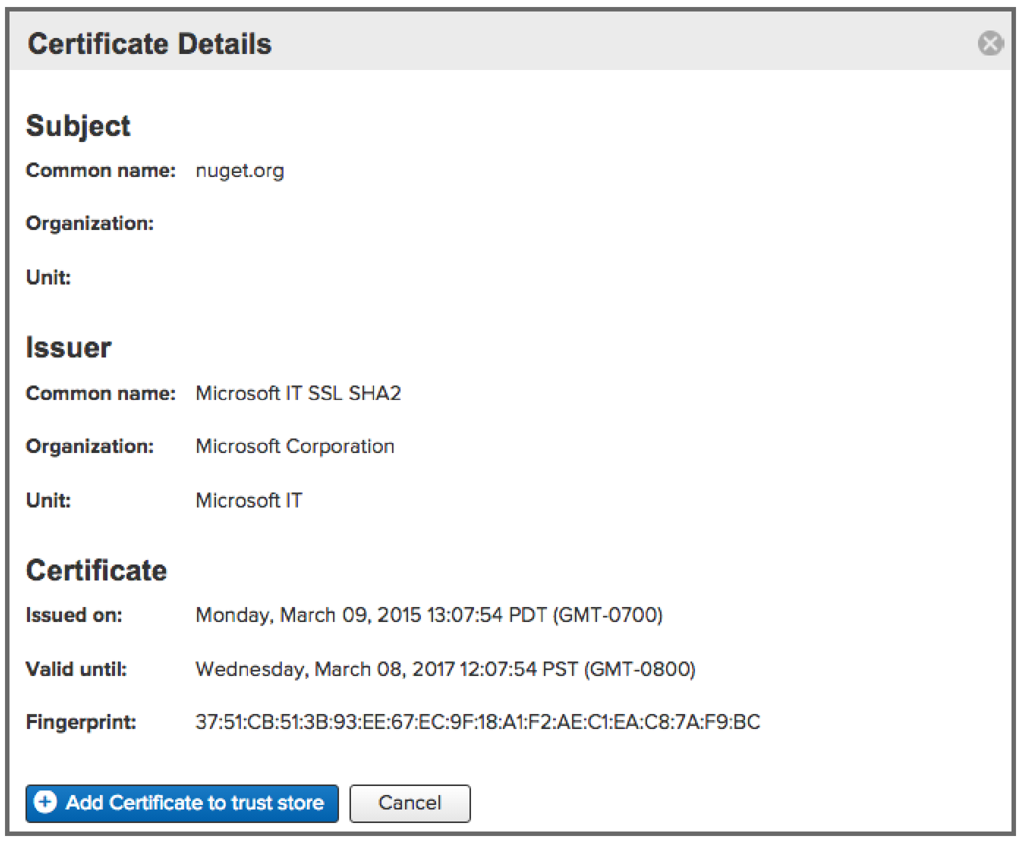
Outbound SSL - Trusting SSL Certificates of Remote Repositories

Available in Nexus Repository OSS and Nexus Repository Pro

When the SSL certificate of a remote proxy repository is not trusted, the repository may be automatically blocked outbound requests fail with a message similar to PKIX path building failed.

The Proxy configuration for each proxy repository documented in Managing Repositories and Repository Groups includes a section titled *Use the Nexus truststore*. It allows you to manage the SSL certificate of the remote repository and solves these problems. It is only displayed, if the remote storage uses a HTTPS URL.

The *View certificate* button triggers the display of the SSL Certificate Details dialog. An example is shown in *Figure 6.16, “Certificate Details Dialog to Add an SSL to the Nexus Truststore”*.


Certificate Details

Subject
Common name: nuget.org
Organization:
Unit:

Issuer
Common name: Microsoft IT SSL SHA2
Organization: Microsoft Corporation
Unit: Microsoft IT

Certificate
Issued on: Monday, March 09, 2015 13:07:54 PDT (GMT-0700)
Valid until: Wednesday, March 08, 2017 12:07:54 PST (GMT-0800)
Fingerprint: 37:51:CB:51:3B:93:EE:67:EC:9F:18:A1:F2:AE:C1:EA:C8:7A:F9:BC

Figure 6.16. Certificate Details Dialog to Add an SSL to the Nexus Truststore

specifically also includes usage of self-signed certificates used in your organization. To confirm trust of the remote certificate, click the *Add certificate to truststore* button in the dialog. This feature is analogous to going to the SSL Certificates user interface and using the Load certificate button found there as described in Outbound SSL - Trusting SSL Certificates Globally. If the certificate is already added, the button can undo this operation and will read *Remove certificate from trust store* .

The checkbox labelled Use certificates stored in Nexus to connect to external systems is used to confirm that the repository manager should consult the internal truststore as well as the JVM truststore when confirming trust of the remote repository certificate. Without adding the certificate to the private truststore and enabling the checkbox, the repository will not trust the remote.

The default JVM truststore of the JVM installation used to run the repository manager and the private truststores are merged. The result of this merge is used to decide about the trust of the remote server. The default Java truststore already contains public certificate authority trust certificates. If the remote certificate is signed by one of these authorities, then explicitly trusting the remote certificate will not be needed.

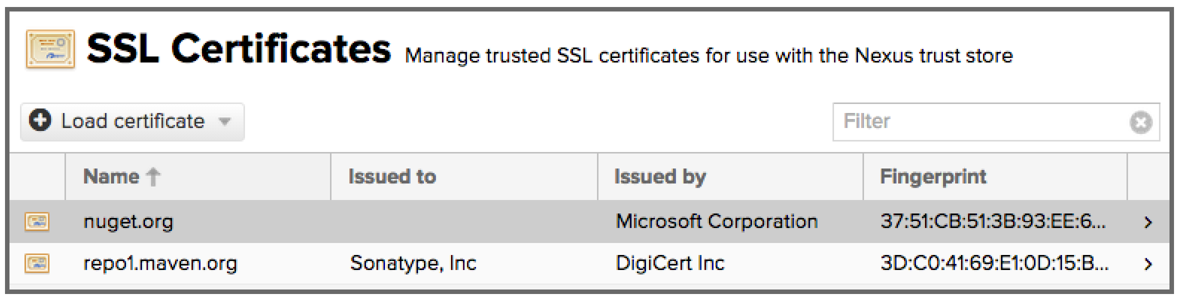
When removing a remote trusted certificate from the truststore, a repository manager restart is required before a repository may become untrusted.

Outbound SSL - Trusting SSL Certificates Globally

Available in Nexus Repository OSS and Nexus Repository Pro

The repository manager allows you to manage trust of all remote SSL certificates in a centralized user interface. Use this interface when you wish to examine all the currently trusted certificates for remote repositories, or manage certificates from secure remotes that are not repositories.

Access the feature view for SSL Certificates administration by selecting the *SSL Certificates* menu items in the *Security* sub-menu in the *Administration* main menu.


SSL Certificates
Manage trusted SSL certificates for use with the Nexus trust store
Load certificate
Filter

Name	Issued to	Issued by	Fingerprint
nuget.org		Microsoft Corporation	37:51:CB:51:3B:93:EE:6...
repo1.maven.org	Sonatype, Inc	DigiCert Inc	3D:C0:41:69:E1:0D:15:B...

Figure 6.17. SSL Certificates Administration

The list shows any certificates that are already trusted. Clicking on an individual row allows you to inspect the certificate. This detail view shows further information about the certificate including *Subject*, *Issuer,* and *Certificate* details. The *Delete certificate* button allows you to remove a certificate from the truststore.

The button *Load certificate* above the list of certificates can be used to add a new certificate to the truststore by loading it directly from a server or using a PEM file representing the certificate.

The common approach is to choose Load from server and enter the full `https://` URL of the remote site, e.g, `https://repo1.maven.org`. The repository manager will connect using HTTPS and use the HTTP proxy server settings if applicable. When the remote is not accessible using `https://`, only enter the host name or IP address, optionally followed by colon and the port number. For example: `example.com:8443`. In this case the repository manager will attempt a direct SSL socket connection to the remote host at the specified port. This allows you to load certificates from SMTP or LDAP servers, if you use the correct port.

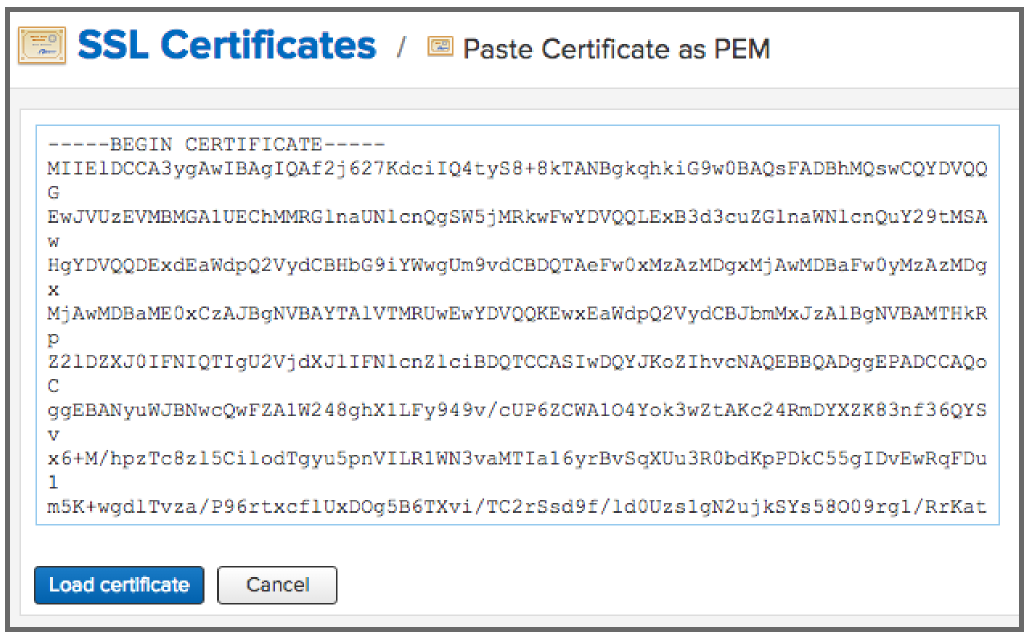
Alternatively you can choose the *Paste PEM* option to configure trust of a remote certificate. Copy and paste the Base64 encoded X.509 DER certificate to trust. This text must be enclosed between lines containing `-----BEGIN CERTIFICATE-----`

and `-----END CERTIFICATE-----` .

Typically this file is supplied to you by the certificate owner. An example method to get the encoded X.509 certificate into a file on the command line using `keytool` is:

```
keytool -printcert -rfc -sslserver repo1.maven.org > repo1.pem
```

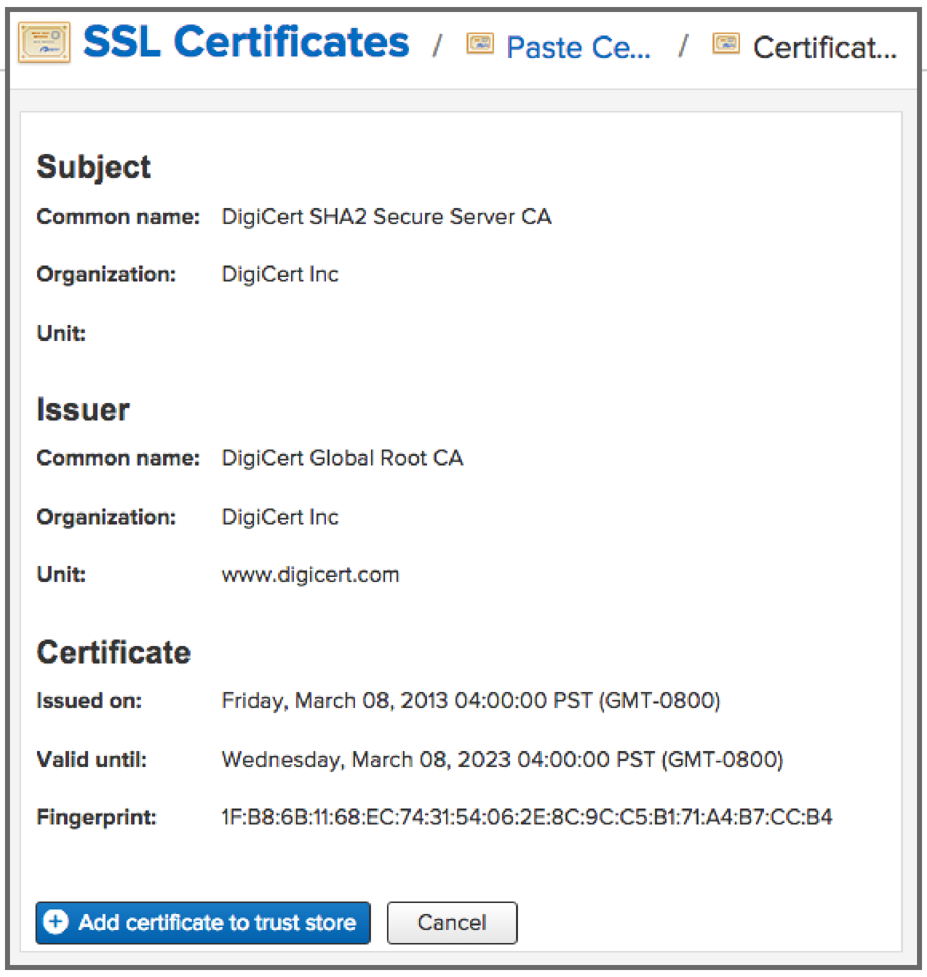
The resulting `repo1.pem` file contains the encoded certificate text that you can cut and paste into the dialog in the user interface. An example of inserting such a certificate is shown in *Figure 6.18, “Providing a Certificate in PEM Format”*.



```
-----BEGIN CERTIFICATE-----
MIIElDCCA3ygAwIBAgIQAf2j627KdciIQ4tyS8+8kTANBgkqhkiG9w0BAQsFADBhMQswCQYDVQQ
G
EwJVUzEVMBMGA1UEChMMRGlnaUNlcnQgSW5jMRkwFwYDVQQLExB3d3cuZGlnaWNlcnQuY29tMSA
w
HgYDVQQDExdEaWdpQ2VydCBHbG9iYWwgUm9vdCBDQTAeFw0xMzAzMDgxMjAwMDBaFw0yMzAzMDg
x
MjAwMDBaME0xCzAJBgNVBAYTAlVTMRUwEwYDVQQKEwxEaWdpQ2VydCBJbmMxJzAlBgNVBAMTHkR
p
Z2lDZXJ0IFNIQTIgU2VjdXJlIFNlcnZlciBDQTCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQo
C
ggEBANyuWJBNwcQwFZA1W248ghX1LFy949v/cUP6ZCWA1O4Yok3wZtAKc24RmDYXZK83nf36QYS
v
x6+M/hpzTc8zl5CilodTgyu5pnVILR1WN3vaMTIa16yrBvSqXUu3R0bdKpPDkC55gIDvEwRqFDu
1
m5K+wgdlTvza/P96rtxcflUxDOg5B6TXvi/TC2rSsd9f/ld0Uzs1gN2ujkSYs58O09rg1/RrKat
```

Figure 6.18. Providing a Certificate in PEM Format

If the repository manager can successfully retrieve the remote certificate or decode the pasted certificate, the details will be shown allowing you to confirm details as shown in *Figure 6.19, “Certificate Details Displayed after Successful Retrieval or Parsing”*. Please review the displayed information carefully before clicking *Add Certificate* to establish the truststore addition.


SSL Certificates / Paste Ce... / Certificat...

Subject

Common name: DigiCert SHA2 Secure Server CA

Organization: DigiCert Inc

Unit:

Issuer

Common name: DigiCert Global Root CA

Organization: DigiCert Inc

Unit: www.digicert.com

Certificate

Issued on: Friday, March 08, 2013 04:00:00 PST (GMT-0800)

Valid until: Wednesday, March 08, 2023 04:00:00 PST (GMT-0800)

Fingerprint: 1F:B8:6B:11:68:EC:74:31:54:06:2E:8C:9C:C5:B1:71:A4:B7:CC:B4

Add certificate to trust store
Cancel

Figure 6.19. Certificate Details Displayed after Successful Retrieval or Parsing

In some organizations, all of the remote sites are accessed through a globally configured proxy server which rewrites every SSL certificate. This single proxy server is acting as a private certificate authority. In this case, you can follow special instructions for trusting the proxy server root certificate , which can greatly simplify your certificate management duties.

Outbound SSL - Trusting SSL Certificates Using Keytool

Available in Nexus Repository OSS and Nexus Repository Pro

Managing trusted SSL certificates from the command line using `keytool` and system properties is an alternative and more complex option than using the SSL certificate management features of the repository manager.

Before you begin the process of trusting a certificate from the command line you will need:

- a basic understanding of SSL certificate technology and how the Java VM implements this feature
- command line access to the host operating system and the `keytool` program
- network access to the remote SSL server you want to trust from the host running the repository manager. This must include any HTTP proxy server connection details.

If you are connecting to servers that have certificates which are not signed by a public CA, you will need to complete these steps:

1. Copy the default JVM truststore file `$JAVA_HOME/jre/lib/security/cacerts` to `$data-dir/custom-truststore.jks` for editing. Ensure the file permissions allow read for the repository manager user.
2. Import additional trusted certificates into the copied truststore file.
3. Configure JSSE system properties for the repository manager process so that the custom truststore is consulted instead of the default file.

Some common commands to manually trust remote certificates can be found in our SSL Certificate Guide.

After you have imported your trusted certificates into a truststore file, you can add the JVM parameters configuring the truststore file location and password as separate configuration lines into the file `$install-dir/bin/nexus.vmoptions`

```
-Djavax.net.ssl.trustStore=<absolute_path_to_custom_truststore_file>
-Djavax.net.ssl.trustStorePassword=<truststore_password>
```

Once you have added the properties shown above, restart the repository manager and attempt to proxy a remote repository using the imported certificate. The repository manager will automatically register the certificates in the truststore file as trusted.

Inbound SSL - Configuring to Serve Content via HTTPS

Available in Nexus Repository OSS and Nexus Repository Pro

Providing access to the user interface and content via HTTPS is a best practice.

You have two options:

- Use a separate reverse proxy server in front of the repository manager to manage HTTPS
- Configure the repository manager itself to serve HTTPS directly

Using A Reverse Proxy Server

A common approach is to access the repository manager through a dedicated server which answers HTTPS requests on behalf of the repository manager - these servers are called reverse proxies or SSL/TLS terminators. Subsequently requests are forwarded to the repository manager via HTTP and responses received via HTTP are then sent back to the requestor via HTTPS the need to work with a custom JVM keystore. The reverse proxy could already be in place for other systems in your network. Common reverse proxy choices are Apache httpd, nginx, Eclipse Jetty or even dedicated hardware appliances. All of them can be configured to serve SSL content, and there is a large amount of reference material available online.

Serving SSL Directly

The second approach is to use the Eclipse Jetty instance that is distributed with the repository manager to accept HTTPS connections.

How to Enable the HTTPS Connector

1. Create a Java keystore file at `$install-dir/etc/ssl/keystore.jks` which contains the Jetty SSL certificate to use. Instructions are available on the Eclipse Jetty documentation site.
2. Edit `$data-dir/etc/nexus.properties`. Add a property on a new line `application-port-ssl=8443`. Change `8443` to be your preferred port on which to expose the HTTPS connector.
3. Edit `$data-dir/etc/nexus.properties`. Change the `nexus-args` property comma delimited value to include `${jetty.etc}/jetty-https.xml`. Save the file.
4. Edit the `$install-dir/etc/jetty-https.xml` and set your keystore password appropriately. Note that if you use "password" as your keystore password this step is not necessary.
5. Restart the repository manager. Verify HTTPS connections can be established.
6. Update the `Base URL` to use `https` in your repository manager configuration using the Base URL capability.

This configuration process is available as a video demonstration.

How to Redirect All Plain HTTP Requests to HTTPS

Some organizations need to remind their users that Nexus Repository Manager should only be used over HTTPS and redirecting HTTP requests to HTTPS can help. Do the following:

1. Follow all the steps under How to Enable the HTTPS Connector. Make sure the nexus-args property value still includes the reference to `${jetty.etc}/jetty-http.xml`
2. Edit `$data-dir/etc/nexus.properties`. Change the nexus-args property comma delimited value to include `${jetty.etc}/jetty-http-redirect-to-https.xml`. Save the file.
3. Restart the repository manager. Verify all plain HTTP requests get redirected to the equivalent HTTPS URL.

Redirecting HTTP requests is not recommended because it introduces implied security and creates increased network latency. Clients which send Basic Authorization headers preemptively may unintentionally expose credentials in plain text.

How to Disable the HTTP Connector

1. Edit `$data-dir/etc/nexus.properties`. Change the `nexus-args` property comma delimited value to not include `${jetty.etc}/jetty-http.xml`. Save the file.
2. Restart the repository manager. Verify plain HTTP requests are no longer serviced.

No labels