

# Network Box SSL Proxy Certificate Installation Guide

Windows and browsers • macOS • iPhone and iPad • Android  
Linux • ChromeOS • Java • Managed deployment • Developer tools

Updated 5 October 2026

Use this guide to download and trust the certificate authority (CA) certificate for your Network Box SSL Proxy deployment. For traffic selected for HTTPS inspection, Network Box inspects and re-encrypts the connection. Your device or application must trust the proxy CA to validate the certificates presented by the proxy.

HTTPS uses TLS; this guide retains the product name **SSL Proxy**. IT separately configures proxy access, routing, and inspection policy.

## Before you begin

- Connect to the network served by your Network Box. **Download the CA from the Network Box GUI on page 2**, selecting the format listed for your platform on page 3.
- Confirm the expected certificate name, issuer, and fingerprint with the administrator. When comparing fingerprints, use the same hash algorithm. Trusting a root CA authorizes certificates issued under that CA.
- Computer-wide installation needs administrator permission. Mobile devices may require a passcode or screen lock. For managed devices, follow IT policy; manual installation may be restricted.

## Choose your platform

| Section                           | Page | Section                           | Page  |
|-----------------------------------|------|-----------------------------------|-------|
| GUI download and format selection | 2-3  | Linux                             | 9     |
| Windows 11: Edge and Chrome       | 4    | Managed ChromeOS                  | 10    |
| macOS: Safari, Chrome and Edge    | 5    | Java applications                 | 11    |
| Firefox                           | 6    | Group Policy, Intune and MDM      | 12    |
| iPhone and iPad                   | 7    | curl, Git and Python              | 13    |
| Android                           | 8    | Node.js and containers            | 14    |
| Verify and troubleshoot           | 15   | References and screenshot credits | 16-17 |

Menu labels can vary by version, device manufacturer, and management policy. Use the certificate for your actual deployment; sample names and filenames in this guide are examples.

# Download from the Network Box GUI

The Network Box GUI provides the proxy CA from its login screen. The public Network Box instructions show the following addresses and download controls. Use the GUI address approved for your deployment. [17]

1. While connected to the network served by Network Box, open the approved GUI address. The standard addresses shown in Network Box documentation are **https://my.network-box.com:4243** and **http://my.network-box.com:4242**. Prefer the approved HTTPS address when it is already trusted.
2. On the GUI **login screen**, select the green **SSL ribbon** shown in Figure 1. The illustrated download control is available before signing in.
3. In **Network Box SSL Certificates**, select **DOWNLOAD** next to the format recommended for your platform on page 3: **DER** or **PEM**.
4. Save the file and note its location. Confirm its identity and fingerprint with IT, then follow the installation section for your platform.

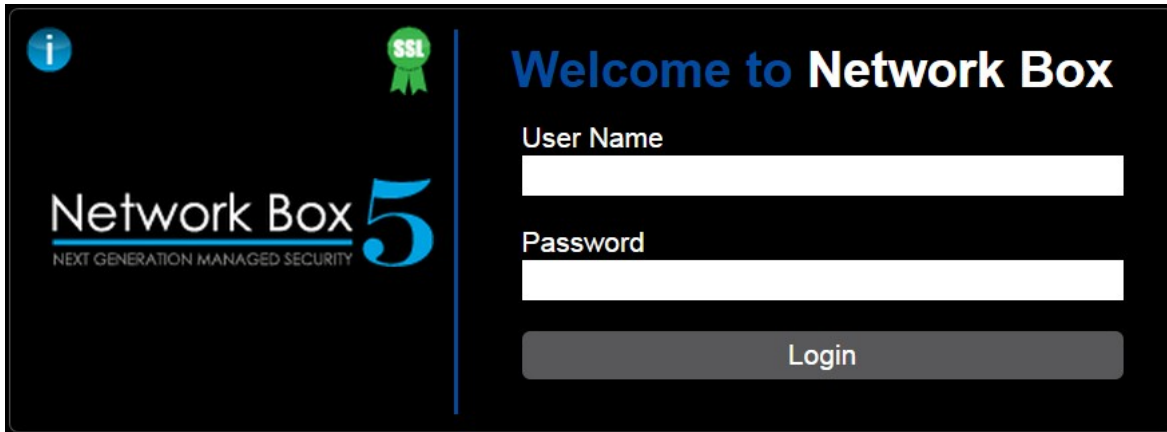


Figure 1. Select the green SSL ribbon on the login screen. Genuine Network Box GUI screenshot from the published certificate guide. [17]

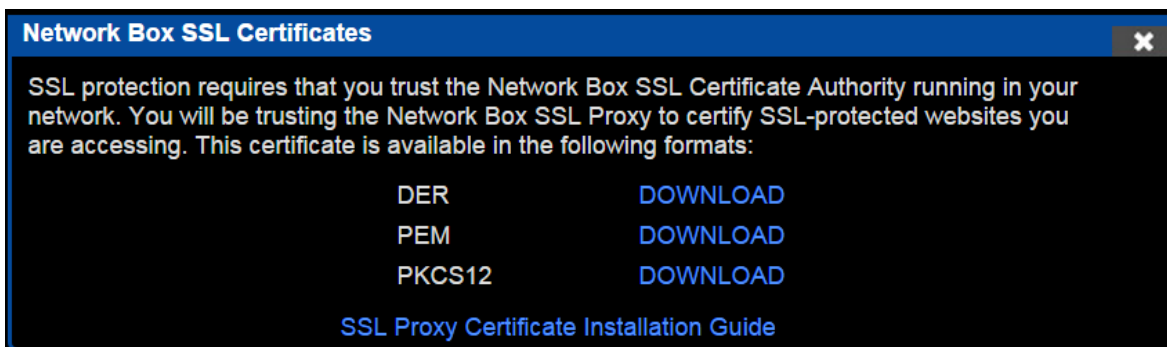


Figure 2. Select DOWNLOAD beside the recommended format. The three GUI options are DER, PEM and PKCS12. [17]

If HTTPS access is blocked by an initial trust warning, ask IT for the approved download route. IT may direct you to the internal HTTP address above; confirm the downloaded CA fingerprint through a trusted channel before installing it. If the address or SSL ribbon is unavailable, ask IT or Network Box support to confirm the GUI address and access policy.

# Choose the certificate format

Select the following option in the **Network Box SSL Certificates** dialog. These are the recommended downloads for the procedures in this guide; some platforms accept both DER and PEM. [17-21]

| Platform or tool                       | GUI option | Installation destination                | Page  |
|----------------------------------------|------------|-----------------------------------------|-------|
| Windows 11: Edge and Chrome            | DER        | Local Computer root store               | 4     |
| macOS: Safari, Chrome and Edge         | DER        | System keychain, then trust             | 5     |
| Firefox manual import                  | DER        | Firefox Authorities for each profile    | 6     |
| iPhone and iPad                        | DER        | Certificate profile, then full trust    | 7     |
| Android                                | DER        | CA certificate / user credentials       | 8     |
| Linux                                  | PEM        | System trust store; check browser trust | 9     |
| Managed ChromeOS                       | PEM        | Google Admin console CA deployment      | 10    |
| Java applications                      | PEM        | The application runtime truststore      | 11    |
| Windows Group Policy / Intune          | DER        | Computer root store via policy          | 12    |
| curl, Git, Python, Node.js, containers | PEM        | Tool CA bundle, runtime or image        | 13-14 |

## What the three options mean

**DER** is a binary X.509 certificate. **PEM** contains the same public certificate as text, normally between BEGIN CERTIFICATE and END CERTIFICATE markers. **PKCS12** is a container format, commonly .p12 or .pfx, which can also contain keys. The CA trust procedures here use DER or PEM and do not require PKCS12 or a CA private key.

For ChromeOS administration, choose **PEM**: Google Admin does not accept DER-encoded CA uploads. Java keytool accepts both encodings; PEM is recommended here for consistency with the published Network Box Java download instructions. [17-19]

## Filenames and certificate identity

The GUI filename can vary. This guide calls a DER download **proxy-ca.cer** and a PEM download **proxy-ca.pem**. A .cer or .crt suffix alone does not identify the encoding. Renaming a file changes its name, not its encoding. Use **All Files** in an import dialog if the download is hidden by its file filter.

For Linux and container commands, the PEM file is copied under a **.crt** name where required. Before trusting either encoding, confirm the expected CA name, issuer, validity dates, and **SHA-256 fingerprint** with IT. Compare the certificate fingerprint, rather than a hash of the downloaded file: DER and PEM file hashes differ even when they contain the same certificate.

# Windows 11: Edge and Chrome

Install the proxy CA in the **Local Computer** root store for Windows 11 or Windows Server. Edge and Chrome recognize locally installed roots; applications with separate trust stores may need additional configuration. [1-3, 15]

1. Open **Start**, search for **Manage computer certificates**, and open it. Alternatively, press **Windows + R**, enter **certlm.msc**, and press **Enter**. Approve elevation or provide administrator credentials when requested.
2. Confirm that the console shows **Certificates (Local Computer)**. Expand **Trusted Root Certification Authorities**, then select **Certificates**.
3. Right-click **Certificates** and select **All Tasks > Import**. Select **Next** in the Certificate Import Wizard.
4. Select **Browse**, choose the **DER** proxy CA downloaded from the Network Box GUI, and select **Open > Next**. If the file is hidden by the filter, choose **All Files**.
5. Choose **Place all certificates in the following store**. Confirm that the store is **Trusted Root Certification Authorities**, then select **Next**.
6. Review the summary and select **Finish**. If a trust confirmation appears, verify the certificate identity before accepting. Select **OK** when the import succeeds.
7. Find the certificate in the store and open it. Check its identity, validity dates, and fingerprint against the information supplied by your administrator. Close and reopen Edge or Chrome, then complete the check on page 15.

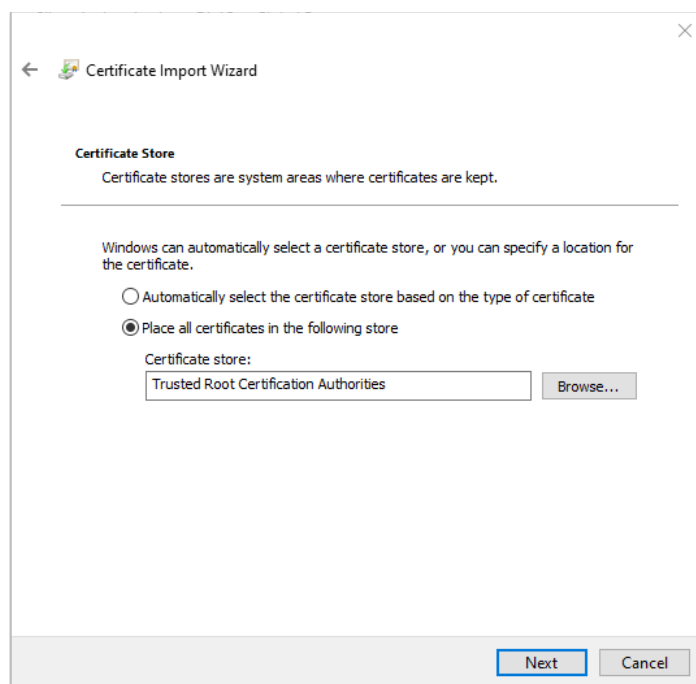


Figure 3. Select Trusted Root Certification Authorities. Cropped reference screenshot from Keytos documentation; Windows styling may differ. [13]

**Store scope matters.** certmgr.msc opens the Current User store; certlm.msc opens the Local Computer store. If you cannot write to the computer store, ask IT to install or deploy the certificate. [1]

## macOS

For Safari, Chrome, Edge, and applications that honor macOS certificate trust, add the proxy CA to the **System** keychain and then set its trust policy. Importing the certificate alone may leave it untrusted. [2-5]

1. Press **Command + Space**, search for **Keychain Access**, and open it. If macOS offers to open the Passwords app, continue to **Keychain Access**.
2. Select the **System** keychain in the sidebar. Use **System**, rather than **System Roots**.
3. Drag the **DER** certificate downloaded from the Network Box GUI into Keychain Access, or choose **File > Import Items** and select it. Enter administrator credentials if requested.
4. Locate and double-click the imported proxy CA certificate. Check that its identity matches the certificate supplied by your administrator.
5. Expand **Trust**. Set **When using this certificate** to **Always Trust**.
6. Close the certificate window and authenticate when requested to save the change. Confirm that the certificate is now trusted. Quit and reopen the browser, then complete the check on page 15.

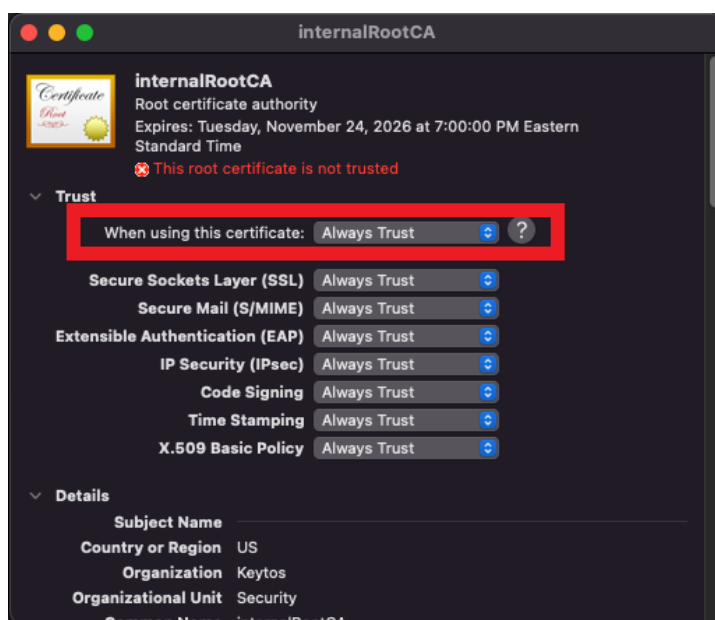


Figure 4. Save the trust change by closing this window and authenticating. internalRootCA is a sample certificate name in the source screenshot. Select your own proxy CA. Cropped Keytos reference screenshot. [13]

The screenshot shows the trust selection before it has been saved; the red message can remain until you close the window and authenticate. The certificate name and appearance on your Mac may differ.

**Managed Macs.** IT may deploy certificate trust through device management. Follow that policy if manual trust changes are unavailable.

# Firefox

Current Firefox on Windows, macOS, and Android normally recognizes third-party root CAs installed in the operating system. A separate Firefox import is therefore usually unnecessary. Install the certificate using the appropriate platform section first. [6]

## Check operating system trust on desktop Firefox

1. Open **Settings > Privacy & Security**, then find **Certificates**.
2. Confirm that **Allow Firefox to automatically trust third-party root certificates you install** is enabled, unless your organization requires another method.
3. Restart Firefox and test an HTTPS site selected for inspection. A certificate trusted through operating system integration may not appear as a separately imported entry in Firefox's certificate manager.

If the control is disabled or locked, ask your IT administrator to check Firefox policy. Administrators can manage **ImportEnterpriseRoots** or the **security.enterprise\_roots.enabled** preference. [6]

## Manual import when required by your administrator

1. On desktop Firefox, open **Settings > Privacy & Security > Certificates > View Certificates**.
2. Select the **Authorities** tab, then **Import**. Choose the **DER** proxy CA downloaded from the Network Box GUI. Firefox can also import the PEM certificate downloaded for Linux.
3. Enable **Trust this CA to identify websites**, then select **OK**. Website trust is the permission needed for HTTPS inspection.
4. Restart Firefox and test again. Repeat this import for each Firefox profile that requires it.

These certificate manager steps are a fallback for an approved manual deployment. Linux integration depends on the distribution and Firefox package; if system trust is not picked up, use the manual import or IT's Firefox certificate policy. [6, 12, 20, 21]

## Firefox on mobile devices

On Android, install the CA using page 8 and test Firefox. Desktop certificate manager steps do not apply to mobile Firefox. On iPhone or iPad, use the certificate profile and full-trust procedure on page 7.

## Managed Firefox

IT can deploy operating system trust and manage the Firefox policy that enables it. Use page 12 for Windows deployment, then verify the affected Firefox profile. Policy restrictions and application-specific certificate behavior still apply.

# iPhone and iPad

For a manually installed root certificate, there are **two required stages**: install the certificate profile, then enable full trust for SSL/TLS. A successful profile installation alone does not complete the process. [7, 8, 16]

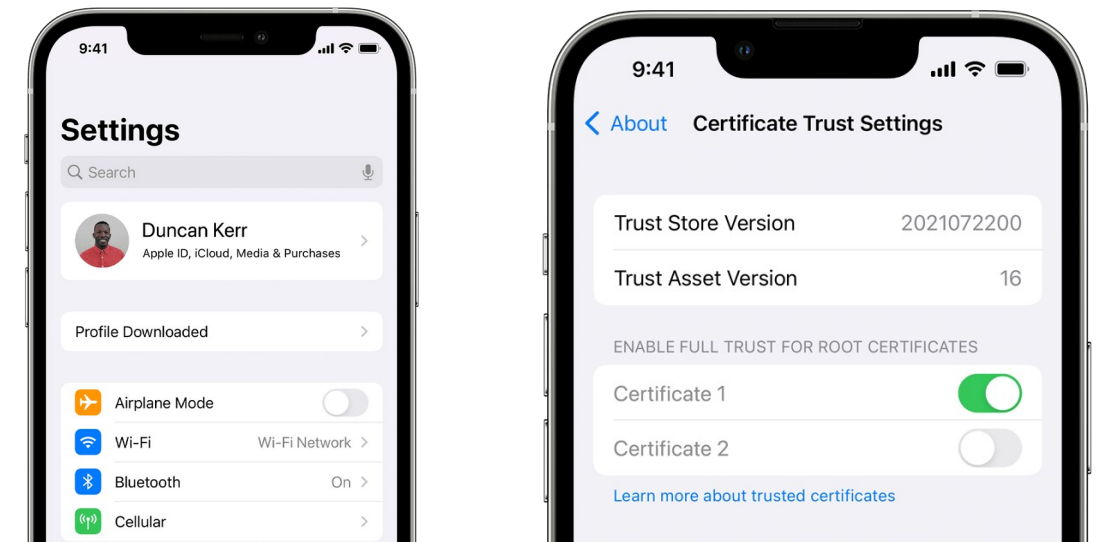
## Install the certificate profile

1. Using Safari, download the **DER** certificate from the Network Box GUI as shown on page 2. Open the download if needed and select **Allow** when prompted to download a profile.
2. Open **Settings** and select **Profile Downloaded**. If needed, check **Settings > General > VPN & Device Management** for the downloaded profile.
3. Open the profile, select **Install**, enter your device passcode when requested, and complete the installation prompts. Confirm that the profile contains the expected proxy CA, then select **Done**.

Install a downloaded profile promptly. Apple deletes a pending profile if it is not installed within eight minutes. If Stolen Device Protection blocks installation, use a familiar location or ask your administrator for assistance. [8]

## Enable full trust

1. Open **Settings > General > About > Certificate Trust Settings**.
2. Under **Enable full trust for root certificates**, enable the proxy CA supplied for your deployment. Accept the confirmation after checking its identity.
3. Confirm that the switch remains on, then reopen Safari or the affected browser and complete the check on page 15.



Figures 5 and 6. Profile installation and full trust are separate actions. Apple Support reference screenshots use iOS 15 styling and sample certificate names; the current steps are given above. [7, 8]

**Managed devices.** Certificates installed by MDM, Apple Configurator, or an MDM enrollment profile are automatically trusted for SSL. A missing or locked manual trust switch may therefore be expected; ask IT to verify the deployment. [7]

# Android

Install the proxy certificate as a **CA certificate** in Settings. Opening a downloaded file alone is not a reliable installation method on current Android versions. Menu names vary by device manufacturer and Android release. [9, 12]

1. Download the **DER** proxy CA from the Network Box GUI and save it on the device, for example in **Downloads**.
2. Open **Settings** and find the certificate installation controls. On recent Pixel devices, start with **Security & privacy > More security settings > Encryption & credentials**. Other devices may use **Security**, **More security & privacy**, or **Other security settings**.
3. Select **Install a certificate > CA certificate**. For this proxy CA, choose the **CA certificate** option. The **Wi-Fi certificate** and **VPN & app user certificate** options serve different purposes.
4. Review the system notice, continue with **Install anyway** if shown, and authenticate with your PIN, pattern, or password. Select the supplied certificate file and complete installation. Enter a recognizable label if prompted.
5. Return to the security settings, open **Trusted credentials > User** or the equivalent user CA list, and confirm that the expected proxy CA is present.
6. Close and reopen the browser. Complete the check on page 15, and test any business apps that must work through inspection.

If you cannot find the menu, search Settings for **CA certificate**, **Install a certificate**, or **Trusted credentials**. If management policy blocks installation, ask IT to deploy it.

## Application trust is separate from installation

By default, apps targeting Android 7.0 or later trust system CAs and do not automatically trust user-added CAs. An app can explicitly allow user certificates through its network security configuration. Apps with certificate pinning may reject inspection even when the CA is installed correctly. [14]

Successful browsing does not prove that every app will accept the proxy CA. Ask IT to investigate an affected app and determine whether its trust configuration or an approved inspection exception is needed. For a work profile, the certificate must be deployed into the profile used by the app.

## If Android reports the wrong certificate type

A message such as **This file cannot be used as a VPN & app user certificate** usually means the wrong installation option was chosen for a CA. Return to the installation menu and select **CA certificate**. If it still fails, ask IT to check the file format and CA certificate properties. [12]

# Linux

Download **PEM** from the Network Box GUI. The commands below install the CA for applications using the distribution's system trust store. They require administrator access and the **ca-certificates** package. Use your approved package source if that package is missing. [20, 21]

## Check the downloaded certificate

In a terminal, change to the download directory. The examples use **proxy-ca.pem**; substitute the actual filename. Inspect its identity, validity dates, and SHA-256 fingerprint, and compare them with IT's expected values. [27]

```
openssl x509 -in proxy-ca.pem -noout -subject -issuer -dates
openssl x509 -in proxy-ca.pem -noout -fingerprint -sha256
```

## Ubuntu and Debian

Copy the PEM certificate with a **.crt** suffix into the local CA directory, then rebuild the system trust store. Each **.crt** file should contain one certificate. [20]

```
sudo cp proxy-ca.pem /usr/local/share/ca-certificates/networkbox-proxy.crt
sudo update-ca-certificates
```

Check that the update reports the added certificate and that **/etc/ssl/certs/networkbox-proxy.pem** is present. Test an inspected HTTPS site with the application that needs access.

## RHEL, Rocky Linux and Fedora

Use the shared CA trust directory and refresh its generated stores. This is the RHEL-family system trust procedure; confirm that your distribution provides these paths and commands. [21]

```
sudo cp proxy-ca.pem /etc/pki/ca-trust/source/anchors/networkbox-proxy.crt
sudo update-ca-trust extract
trust list --filter=ca-anchors
```

Confirm that the expected Network Box proxy CA appears as a trust anchor, then restart the affected application and test.

## Browsers, isolated apps and WSL

Browser trust can depend on the distribution, package, or sandbox. Snap-packaged applications may not use the host CA update. If Firefox still reports a trust error, use the approved manual import on page 6. Ask IT to deploy the appropriate CA policy for other browsers. [20, 21]

Treat a WSL distribution or Linux virtual machine as a separate Linux environment: install the CA inside the environment used by the tool. Java, Python, Node.js, and containers may also need their own trust configuration; see pages 11 and 13-14.

# ChromeOS: managed Chromebooks

For organizational Chromebooks, deploy the CA through **Google Admin**. Download **PEM** from the Network Box GUI. Google Admin accepts a single PEM-encoded certificate per upload and rejects DER-encoded certificates. [18]

## Deploy the CA in Google Admin

1. Sign in to **admin.google.com** with an administrator account that has the required device/network management privilege.
2. Open **Devices > Networks > Certificates**. Select the organizational unit that should receive the proxy CA.
3. Select **Create certificate** and enter a recognizable certificate name.
4. Select **Upload**, choose the **PEM** proxy CA downloaded from Network Box, and select **Open**. Upload a file containing exactly one certificate.
5. Under **Certificate Authority**, select the intended platforms, including **ChromeOS**, then select **Add**.
6. Allow management policy to reach the target devices, then verify the certificate and an inspected HTTPS connection.

## Verify on the Chromebook

1. Open Chrome settings at **chrome://settings**.
2. Go to **Privacy and security > Security** and open **Manage certificates**; it may appear under **Advanced**. Search Settings for this control if its location has changed.
3. Find the deployed proxy CA and confirm its identity. Test an HTTPS website selected for inspection as described on page 15.

## Device management and application scope

An organizational policy can restrict manual certificate changes. For an unmanaged Chromebook, use the approved local CA import or enroll the device according to IT policy; the Admin console steps require a managed deployment.

A Linux environment or an application using its own CA store needs separate verification. Use the Linux and runtime sections when those environments access inspected HTTPS services.

# Java applications

Download **PEM** from the Network Box GUI. Modern Java applications commonly validate TLS with a runtime or application truststore. Browser trust and the legacy Java Control Panel are not sufficient checks for the Java runtime your application actually uses. [17, 19]

## Select the application's truststore

Ask the application owner which Java installation and truststore it uses. A bundled runtime or a **javax.net.ssl.trustStore** setting can differ from the Java on your command path. Use that runtime's **keytool** and back up the existing truststore before changing it.

The commands below assume keytool is from the correct runtime. Replace filenames and paths as needed. PEM and DER certificates are both accepted by keytool. [19]

## Inspect the CA before import

```
keytool -printcert -file proxy-ca.pem
```

Compare the displayed SHA-256 fingerprint and certificate identity with IT's expected values.

## Import into the runtime's existing cacerts store

For a runtime using its default **cacerts** store, run with permission to modify that store. Enter the truststore password when prompted, then confirm the import after checking the displayed certificate:

```
keytool -importcert -cacerts -alias networkbox-ssl-proxy -file proxy-ca.pem
```

Confirm the entry:

```
keytool -list -v -cacerts -alias networkbox-ssl-proxy
```

## When the application uses a custom truststore

Use the application's supported certificate management method. If IT directs a keytool import, select its **existing** truststore rather than creating an empty replacement:

```
keytool -importcert -keystore "/path/to/existing-truststore" -alias networkbox-ssl-proxy  
-file proxy-ca.pem
```

Enter that command on one line. Preserve the application's other trusted CAs and its required store type. If the alias already exists, have IT compare the current entry before replacing it.

Restart the application and test an inspected HTTPS request. Runtime upgrades or application reinstalls may replace the truststore; include CA deployment in that application's maintenance process.

# Managed certificate deployment

Use centralized deployment for fleets of devices. Deploy the **public proxy CA certificate** and confirm that the policy reaches the store or profile used by the application. [7, 10, 11]

## Active Directory Group Policy: Windows

1. Download **DER** from the Network Box GUI. On a domain administration workstation, open **Group Policy Management**.
2. Create or select a GPO linked to the organizational unit containing the target **computers**. Edit the GPO.
3. Open **Computer Configuration > Policies > Windows Settings > Security Settings > Public Key Policies > Trusted Root Certification Authorities**.
4. Right-click **Trusted Root Certification Authorities**, select **Import**, and choose the downloaded DER CA. Complete the wizard with that root store as the destination.
5. Apply the policy and restart a pilot device as required. Use **gpupdate /force** for a policy refresh where appropriate. In **certlm.msc**, confirm the CA is in the computer root store, then test Edge, Chrome, and required applications.

A proxy root CA belongs in **Trusted Root Certification Authorities**. **Trusted Publishers** is not the destination for HTTPS proxy root trust. If IT supplies a separate intermediate CA, deploy that intermediate to **Intermediate Certification Authorities**; do not duplicate the root there. [1, 10]

## Microsoft Intune: Windows

1. Download **DER**. If necessary, save or rename the DER file with a **.cer** extension; a PEM file must be converted, rather than just renamed.
2. In Intune, create a Windows configuration profile using the **Trusted certificate** template. Upload the public CA **.cer** file.
3. Set the destination to **Computer certificate store - Root**. Assign the profile to the intended groups, then check deployment status and verify the CA on a pilot device.

## Apple, Android and ChromeOS management

On macOS, iPhone, and iPad, use an approved MDM certificate payload or Apple Configurator workflow. For iPhone and iPad, MDM or Configurator installation establishes SSL trust without the separate manual full-trust switch. [7]

For Android Enterprise, deploy the CA into the device or work profile used by the application and verify that the application accepts it. User CA restrictions and certificate pinning can still prevent inspection. [14]

For managed Chromebooks, follow the Google Admin **PEM** deployment procedure on page 10.

## Pilot the rollout and CA replacement

Verify each OS and important application before broad deployment. When replacing the proxy CA, distribute the new CA before changing the service, then retire the old CA through the policy that installed it once it is no longer needed.

# curl, Git and Python

Download **PEM** for tools that use a file-based CA store. Replace sample paths and **https://example.com** with an HTTPS test site selected for inspection. A custom CA bundle should retain the tool's normal trusted public CAs and add the proxy CA, so excluded connections still validate.

## curl

curl's trust source depends on its TLS backend. **curl -V** shows the build details; Schannel builds normally use Windows certificate trust. For a file-based backend, use the updated system store or an approved PEM CA bundle. [22]

```
curl --cacert /absolute/path/ca-bundle.pem https://example.com
```

The bundle must include the proxy CA. A per-transfer **--cacert** option or **CURL\_CA\_BUNDLE** can select the bundle; the relevant backend determines support. Keep certificate verification enabled.

## Git over HTTPS

On Git for Windows builds supporting Schannel, **http.sslBackend=schannel** can use the Windows root store installed on page 4. For an OpenSSL/file-based setup, point **http.sslCAInfo** at an approved PEM CA bundle. [23]

```
git config --global http.sslBackend schannel
```

Use the preceding command only for the supported Windows setup. For a file-based setup, use this configuration instead, preserving the bundle's existing public roots:

```
git config --global http.sslCAInfo "/absolute/path/ca-bundle.pem"
```

Test against the actual Git server. Confirm the effective backend and organization policy before changing an existing configuration.

## Python Requests

Requests commonly uses **certifi**, so an OS CA install may not be enough. Configure **verify** or **REQUESTS\_CA\_BUNDLE** with a PEM bundle. This example creates a separate bundle containing certifi's public roots plus the proxy CA. [24]

```
from pathlib import Path
import certifi
import requests

bundle = Path("networkbox-ca-bundle.pem").resolve()
bundle.write_bytes(Path(certifi.where()).read_bytes()
                  + b"\n" + Path("proxy-ca.pem").read_bytes() + b"\n")
response = requests.get("https://example.com",
                        verify=str(bundle), timeout=30)
```

Refresh the separate bundle when certifi or the proxy CA changes. Applications using prepared requests must also apply the environment settings when relying on **REQUESTS\_CA\_BUNDLE**. [24]

# Node.js and containers

## Node.js and tools built on it

Download **PEM**. **NODE\_EXTRA\_CA\_CERTS** adds the proxy CA to Node.js's normal trusted roots. Set it before launching the process; it is read at startup. [25]

Linux or macOS shell:

```
export NODE_EXTRA_CA_CERTS="/absolute/path/proxy-ca.pem"
node app.js
```

Windows PowerShell:

```
$env:NODE_EXTRA_CA_CERTS = "C:\certificates\proxy-ca.pem"
node app.js
```

For a service, configure the variable in its service environment and restart it. An application that explicitly sets a **ca** option can override the usual roots and extra CA file. Node versions supporting **--use-system-ca** can also use operating system trust; confirm the option exists in your deployed runtime. [25]

## Docker and other containers

Host trust and container trust are separate. Install the proxy CA on the host for host-side connections, and in an image for applications running inside it. Rebuild the image to preserve the CA when containers are recreated. [26]

1. Download **PEM** and copy it into your build context as **networkbox-proxy.crt**. This changes the filename; the contents remain PEM.
2. For a Debian/Ubuntu-based image with **ca-certificates** already installed, add the Dockerfile instructions below. For other base distributions, use their supported trust-store procedure.
3. Build a new image, recreate the containers from that image, and test an inspected HTTPS request from inside the application environment.

```
COPY networkbox-proxy.crt /usr/local/share/ca-certificates/
RUN update-ca-certificates
```

If the base image lacks ca-certificates, include its approved package installation before updating the store. A runtime-only change is temporary and is lost when the container is replaced. Java, Python, or Node.js inside a container may still need the runtime settings in this guide. [26]

## Other applications with their own CA settings

API clients, package managers, build runners, and vendor applications can maintain separate CA stores. Follow their supported CA import or bundle setting, usually using **PEM**, and verify the actual application. Ask IT to assess tools that use certificate pinning or do not support an organizational CA.

Across these tools, resolve trust errors by selecting the correct CA store. Keep TLS certificate verification enabled; options that disable verification do not install or validate the proxy CA.

# Verify the installation

Test while connected through the Network Box service, using an HTTPS website your administrator confirms is selected for inspection. A site that is excluded from inspection cannot verify proxy certificate trust.

1. Close and reopen the affected browser, then open the test site. It should load without a certificate warning.
2. Where the browser provides a certificate viewer, open the connection information and inspect the site certificate chain. For a connection confirmed to be inspected, the chain should lead to the proxy CA supplied for your deployment. The CA name can vary by customer.
3. Confirm with the administrator that the connection was actually inspected, and test the business applications you need to use. Browser success alone does not validate applications with separate certificate trust settings.

## Troubleshooting

| Symptom                                          | What to check                                                                                                                                    |
|--------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------|
| Certificate warning in every browser             | Confirm the correct CA, destination store, saved trust setting, and certificate validity. Verify the device date and time.                       |
| Warning only in Firefox                          | Check the operating system trust setting and Firefox policy, or the approved manual import in that Firefox profile.                              |
| iPhone or iPad profile installed but HTTPS fails | Check Certificate Trust Settings. Manual profile installation and full trust are separate steps.                                                 |
| Android browser works but an app fails           | Check whether the app accepts user CAs, uses certificate pinning, or runs in a separate work profile.                                            |
| Browser works but Java or a tool fails           | Check the runtime, CA bundle, container image, or application truststore actually used for that connection.                                      |
| The test site has a public CA issuer             | The connection may be excluded from inspection or using a different route. Ask IT to confirm the inspection policy.                              |
| Import requires a private key or a key password  | Confirm that you have the public proxy CA and selected the CA installation method. A proxy trust deployment does not require the CA private key. |

## When requesting support

Provide the device OS version, browser or app name and version, affected website or service, exact error message, and a screenshot. Include the proxy CA name and fingerprint if available. Your administrator can use these details to check certificate trust and inspection policy.

**Do not bypass a certificate warning to complete this test.** Resolve the trust or inspection issue with IT or Network Box support.

## References: platforms and deployment

Procedures reviewed against published documentation on 5 October 2026. The GUI illustrations show the Network Box controls in the supplied guide. Other reference screenshots illustrate certificate controls; their styling and sample names may differ from your device.

[1] **Microsoft Learn** [Manage certificate stores and understand computer and user scopes](#)

[2] **Microsoft Learn** [Microsoft Edge TLS server certificate verification](#)

[3] **Chromium** [Chrome Root Store frequently asked questions](#)

[4] **Apple Support** [Add certificates to a keychain on Mac](#)

[5] **Apple Support** [Change certificate trust settings on Mac](#)

[6] **Mozilla Support** [Set up certificate authorities in Firefox](#)

[7] **Apple Support** [Trust manually installed certificate profiles](#)

[8] **Apple Support** [Install a configuration profile on iPhone or iPad](#)

[9] **Google Pixel Help** [Add and remove certificates and find credential settings](#)

[10] **Microsoft Learn** [Distribute certificates using Group Policy](#)

[11] **Microsoft Learn** [Trusted root certificate profiles for Intune](#)

[12] **Proxyman documentation** [Android CA installation](#) | [Firefox manual import](#)

[13] **Keytos documentation** [Trust a root certificate in Windows and macOS](#)

[14] **Android Developers** [Network security configuration and application CA trust](#)

[15] **Microsoft Learn** [Import certificates through the Local Computer console](#)

[16] **Apple Support** [Find and manage configuration profiles on iPhone](#)

## References and screenshot credits

[17] **Network Box USA** Original SSLProxyCertificate-IE.pdf, page 1 (GUI illustrations), and SSLProxyCertificate-Java.pdf, page 1 (PEM download instructions).

[18] **Google Chrome Enterprise** [Set up an HTTPS certificate authority for ChromeOS](#)

[19] **Oracle Java documentation** [keytool certificate import and truststore commands](#) | [JSSE runtime truststore selection](#)

[20] **Ubuntu Server documentation** [Install a root CA certificate in the system trust store](#)

[21] **Red Hat documentation** [Use shared system certificates in RHEL 9](#)

[22] **curl documentation** [TLS certificate verification and native or file-based stores](#)

[23] **Git documentation** [HTTPS CA settings and TLS backends](#)

[24] **Python Requests documentation** [CA bundles, verify and environment configuration](#)

[25] **Node.js documentation** [NODE\\_EXTRA\\_CA\\_CERTS and system CA options](#)

[26] **Docker documentation** [CA certificates in hosts, images and containers](#)

[27] **OpenSSL documentation** [x509 certificate inspection and fingerprint options](#)

### Screenshot sources

**Figures 1 and 2:** genuine Network Box GUI images extracted unchanged from page 1 of [17]. They show the SSL ribbon and DER, PEM, and PKCS12 download options.

**Figures 3 and 4:** genuine Windows and macOS images from Keytos [13], cropped to the relevant dialogs. These examples have older styling; the accompanying steps describe the published current method.

**Figures 5 and 6:** genuine Apple Support examples from [8] and [7], published with iOS 15 styling. Follow the current profile installation and full-trust procedure in this guide.

Sample names such as **internalRootCA**, **Certificate 1**, and **Certificate 2** do not identify your deployment's CA. Use the certificate downloaded from your Network Box and identity details confirmed by IT.