**To convert a .pem file to .crt and .key files**

**On Windows:**

* Use the **Certificate Import Wizard** to export certificates in .pfx format:
  1. Open the Microsoft Management Console (MMC).
  2. Add the **Certificates** snap-in.
  3. Import your .crt and .key files.
  4. Right-click the certificate and choose **Export** to .pfx.

**On Linux (Without OpenSSL):**

* A screenshot of a computer

  Description automatically generatedIf openssl is not available, tools like **certbot** or the certificates package on Debian-based systems can help manage formats.

**Step 1: Ensure Windows Is Updated**

1. Open **Windows Update Settings** and ensure your system is up to date.
2. Restart your computer if required after update

**Step 2: Open Command Line with Admin Privileges**

1. Open the **Windows Terminal** or **PowerShell**.
2. Run it with **administrator privileges**:
3. Right-click on the app and select **Run as Administrator**.

**Step 3: Check for Winget (Windows Package Manager)**

1. Verify that the winget tool is installed on your system

winget –version

1. If you see a version number, proceed. If not, [install Winget](https://github.com/microsoft/winget-cli).

**Step 4: Search for OpenSSL**

1. Use Winget to find the OpenSSL package:

winget search openssl

1. Look for a result similar to **"Shining Light Productions OpenSSL"**.

**Step 5: Install OpenSSL**

1. Install the OpenSSL package using Winget

winget install -e --id ShiningLight.OpenSSL.Light

1. Follow the prompts to complete the installation. The package is installed in the default directory (C:\Program Files\OpenSSL-Win64).

A screenshot of a computer

Description automatically generated

**Step 6: Verify OpenSSL Installation**

1. Check the OpenSSL version:

openssl version

1. If you see an error like "OpenSSL is not recognized as an internal or external command," it means the path is not set.

**Step 7: Configure the Path (If Needed)**

1. Add the OpenSSL binary directory to the Windows PATH:

setx PATH "%PATH%;C:\Program Files\OpenSSL-Win64\bin" /M

1. Close and reopen your terminal to apply the updated PATH.
2. Run the following command again to verify OpenSSL:

openssl version

1. You should now see the installed version of OpenSSL.

**Step 8: Generate a Private Key**

1. Create a private key using OpenSSL:

openssl genrsa -out mykey.key 2048

1. This will generate a file named mykey.key containing your private key.

**Step 9: Create a Self-Signed Certificate**

1. Use the private key to generate a self-signed certificate

openssl req -new -x509 -key mykey.key -out mycert.pem -days 365

or:

openssl x509 -outform der -in your-cert.pem -out your-cert.crt

openssl x509 -outform der -in yourPemFilename.pem -out certfileOutName.crt

openssl rsa -in yourPemFilename.pem -out keyfileOutName.key

1. You’ll be prompted to enter information about your certificate (country, state, organization, etc.). You can skip any field by pressing **Enter**.

A computer screen with white text

Description automatically generated

**Step 10: Verify the Certificate**

1. Inspect the details of the generated certificate:

openssl x509 -text -noout -in mycert.pem

1. This will display the certificate’s information, confirming that it was created successfully.

**Common Issues and Fixes**

1. **Error: OpenSSL is not recognized**
2. Ensure the PATH variable includes the OpenSSL binary directory.
3. **Installation issues with Winget**
4. Download OpenSSL manually from Shining Light Productions.

**Step 11: Download the Required SSL Certificate Files**

1. Obtain the SSL certificate files from your Certificate Authority (CA).
2. Save them to a folder on your local machine. These files typically include:
3. The .crt file (certificate file).
4. Other supporting files, such as intermediate and root certificates (if provided).

**Step 12: Install OpenSSL**

1. Download OpenSSL for your operating system:
2. Visit OpenSSL's official website or another trusted source.
3. Select the appropriate version for your system (e.g., 64-bit or 32-bit).
4. Install the software using the default installation options.
5. After installation, open the "OpenSSL Command Prompt" (e.g., Win64OpenSSL).

**Step 13: Prepare the Environment**

1. Navigate to the directory where your SSL files are located:
2. cd path/to/your/certificate/folder
3. Replace path/to/your/certificate/folder with the path to the directory containing your .crt file.

**Step1 4: Convert .crt to .pem Format**

1. To convert the .crt file to .pem format:
2. Run the following command:
3. openssl x509 -in certificate.crt -outform PEM -out certificate.pem
4. Replace certificate.crt with the name of your .crt file.The output file will be certificate.pem.
5. Verify the .pem file is created successfully in the directory.

**Step 15: Convert .crt to .pfx Format**

To convert the .crt file (with the private key) to .pfx:

1. Ensure you have the private key file (private.key). If not, contact your CA.
2. Run the following command:
3. openssl pkcs12 -export -out certificate.pfx -inkey private.key -in certificate.crt
4. Replace private.key with your private key file.
5. Replace certificate.crt with your certificate file.
6. The output file will be certificate.pfx.
7. You will be prompted to create a password to protect the .pfx file. Enter a secure password.

**Step 16: Verify the Conversion**

1. Check that the .pem and .pfx files are in the directory.
2. Use the .pem file for applications requiring PEM format and .pfx for those requiring PKCS#12 format.

**Step 17: Optional: Test the Certificate**

1. Verify the certificate by testing it online or using tools that accept .pem or .pfx formats.
2. For example, you can use SSL testing services to confirm proper installation and functionality.

**Automate with Scripts**

If frequently need to perform this operation, automate it with a simple script.

#!/bin/bash

# Script to convert .crt to .pem and .pfx

CERT\_FILE="certificate.crt"

KEY\_FILE="private.key"

PEM\_OUTPUT="certificate.pem"

PFX\_OUTPUT="certificate.pfx"

# Convert to .pem

openssl x509 -in $CERT\_FILE -outform PEM -out $PEM\_OUTPUT

echo "Converted to PEM format: $PEM\_OUTPUT"

# Convert to .pfx

openssl pkcs12 -export -out $PFX\_OUTPUT -inkey $KEY\_FILE -in $CERT\_FILE

echo "Converted to PFX format: $PFX\_OUTPUT"

* Save the script as convert\_cert.sh.
* Run it:

bash convert\_cert.sh