



Mercurion Systems, Inc.

Information Technology Consulting and Support Services

Phone: 919.266.5957 Fax: 919.217.5769
<http://www.mercurionsystems.com>

CONVERTING CERTIFICATE FORMATS WITH OPENSLL

Products: Stunnel 4.04; OpenSSL (various versions)

Overview

Stunnel uses the PEM (originally Privacy Enhanced Mail, now a standard for encoding certificates) format for its certificates. Windows, on the other hand, typically uses the PFX (PKCS #12) format for storing certificates. To prepare certificates for use with Stunnel, the certificates must be converted to PEM format.

This technical note describes how to use the OpenSSL toolkit to convert certificates from PFX format to PEM format for use with Stunnel.

More Information

Certificate Conversion on Mac OS X and Linux

The following steps can be used to convert a certificate (with private key) in PFX format to a certificate (with an unencrypted private key) in PEM format from a Linux-based or Mac OS X-based system. These steps require the use of OpenSSL, an open-source SSL toolkit. These steps were confirmed using OpenSSL 0.9.7a on Red Hat Linux 9.0 and OpenSSL 0.9.6i on Mac OS X 10.2.8.

1. If the certificate is not already available in PFX format, use the Certificates MMC snap-in to export the certificate and the corresponding private key to a PFX file. Be sure to note the passphrase used to protect the PFX file; it will be needed later. If the certificate (and its corresponding private key) is already available in PFX format and the passphrase for the PFX file is known, proceed to step 2.
2. At a terminal prompt, type “`openssl pkcs12 -in pfxfilename.pfx -out tempfile.pem`”. This will convert the PFX file to a PEM file. The OpenSSL toolkit will prompt for the import passphrase; this will be the passphrase specified in step 1. A PEM passphrase will also be needed; make note of the passphrase used here (it will be needed later).
3. Using a text editor such as vi, split the encrypted RSA private key and the certificate into two separate files. Remove all extra text, leaving only the text between the lines with the dashes. Make note of the filenames; they will be needed later.
4. The RSA key is currently encrypted; this will cause Stunnel to prompt for the private key’s passphrase when Stunnel is launched—thus preventing any sort of automated launch. It will be necessary to decrypt the RSA private key. To decrypt the RSA private key, use the

command “openssl rsa -in *encryptedkey* -out *decryptedkey*” (where *encryptedkey* is the file containing the RSA private key, as separated in step 3, and *decryptedkey* is the file that will contain the decrypted RSA private key). The OpenSSL toolkit will prompt for the RSA key passphrase; this will be the passphrase specified in step 2.

5. To concatenate the decrypted RSA private key and the certificate into a single file that Stunnel can use, use the command “cat *decryptedkey* *certificatefile* > *finalfile.pem*” (where *decryptedkey* is the decrypted RSA key produced in step 4 and *certificatefile* is the file containing only the certificate, as produced in step 3).
6. Edit the concatenated file (*finalfile.pem* from step 5) with a text editor to add a blank line between the decrypted RSA private key and the certificate, and a blank line after the end of the certificate.

At this point, the certificate is ready for Stunnel to use.

Certificate Conversion on Windows 2000/2003

The following steps can be used to convert a certificate (with private key) in PFX format to a certificate (with an unencrypted private key) in PEM format from a Windows 2000/2003-based system. These steps require the use of OpenSSL, an open-source SSL toolkit. These steps were confirmed using OpenSSL 0.9.7c on Windows Server 2003, Standard Edition.

1. If the certificate is not already available in PFX format, use the Certificates MMC snap-in to export the certificate and the corresponding private key to a PFX file. Be sure to note the passphrase used to protect the PFX file; it will be needed later. If the certificate (and its corresponding private key) is already available in PFX format and the passphrase for the PFX file is known, proceed to step 2.
2. From a command line, type “openssl pkcs12 -in *pfxfilename.pfx* -out *tempfile.pem*”. This will convert the PFX file to a PEM file. The OpenSSL toolkit will prompt for the import passphrase; this will be the passphrase specified in step 1. A PEM passphrase will also be needed; make note of the passphrase used here (it will be needed later).
3. Using a text editor such as Notepad, split the encrypted RSA private key and the certificate into two separate files. Remove all extra text, leaving only the text between the lines with the dashes. Make note of the filenames; they will be needed later.
4. The RSA key is currently encrypted; this will cause Stunnel to prompt for the private key’s passphrase when Stunnel is launched—thus preventing any sort of automated launch. It will be necessary to decrypt the RSA private key. To decrypt the RSA private key, use the command “openssl rsa -in *encryptedkey* -out *decryptedkey*” (where *encryptedkey* is the file containing the RSA private key, as separated in step 3, and *decryptedkey* is the file that will contain the decrypted RSA private key). The OpenSSL toolkit will prompt for the RSA key passphrase; this will be the passphrase specified in step 2.
5. To combine the decrypted RSA private key and the certificate into a single file that Stunnel can use, use the command “copy /b *decryptedkey*+*certificatefile* *finalfile.pem*” (where *decryptedkey* is the decrypted RSA key produced in step 4 and *certificatefile* is the file containing only the certificate, as produced in step 3).

6. Edit the combined file (*finalfile.pem* from step 5) with a text editor to add a blank line between the decrypted RSA private key and the certificate, and a blank line after the end of the certificate.

The certificate is now ready for Stunnel to use.

Other Notes

None

Related Articles/Resources

More information about Stunnel (the universal SSL wrapper) can be found at <http://www.stunnel.org>; more information on OpenSSL can be found at <http://www.openssl.org>.

Legal Information

This document was created by Mercurion Systems, Inc., and may be freely distributed as long as it is distributed without modification and in its entirety. Mercurion Systems assumes no liability as a result of using the information contained in this document.