



Mercurion Systems, Inc.

Information Technology Consulting and Support Services

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

ENHANCING EXCHANGE 2000 SERVER WITH STUNNEL

Products: Stunnel 4.04; Exchange 2000 Server, Standard Edition; various Internet e-mail clients

Overview

Microsoft Exchange 2000 Server provides SMTP, POP3, and IMAP functionality that can be used to support remote users. Each of these protocols also provides support for SSL/TLS encryption, using digital certificates, to provide an additional layer of security for these remote users. In fact, using SSL/TLS with these Internet-based protocols is considered a best practice, given that these protocols pass authentication information in clear text between the client and the server.

However, Exchange Server 2000 only supports SMTP over SSL (SMTPS) on TCP port 25, rather than using TCP port 465 (as listed by the Internet Assigned Numbers Authority [IANA]). In most situations, this would not be an issue. However, in some cases it may be necessary to have the ability to run SMTPS on TCP port 465 instead of TCP port 25, perhaps to enforce SSL use by SMTP clients, for consistency with the other e-mail protocols (both POP3 and IMAP operate on different TCP ports when using SSL), or perhaps for interoperability or co-existence with firewalls and other perimeter security devices.

This document describes how to use Stunnel, an open-source utility referred to as the “universal SSL wrapper,” to provide this functionality and enhance the built-in support for SSL that Exchange 2000 Server offers.

More Information

Stunnel (available from <http://www.stunnel.org>) is an open-source utility that provides the ability to encapsulate virtually any type of traffic inside an SSL wrapper. Stunnel does not provide its own encryption functionality, relying instead upon a library such as OpenSSL (available from <http://www.openssl.org>). The sections below provide more information on how to use Stunnel to enhance Exchange 2000 Server's native SSL support.

Obtain SSL Certificates

In order to use SSL to secure POP3, IMAP, and SMTP, one or more SSL certificates will be needed. These certificates may be obtained from a commercial certificate authority, or from an internal certificate authority (such as one created using Certificate Services available in Windows 2000 Server or Windows Server 2003). This technical note was written from the perspective of having the necessary SSL certificates in PFX format (including private key). The passphrase for the PFX file (used to protect the file, since it contains the certificate's private key)

must be known. It does not matter if the certificates were obtained from an internal certificate authority or a commercial certificate authority.

Configuring Exchange 2000 Server for POP3S or IMAPS

As mentioned earlier, Exchange 2000 Server has built-in support for SSL/TLS with POP3, IMAP, and SMTP. POP3 and IMAP operate on TCP ports 110 and 143, respectively, when SSL/TLS is not in use. When SSL/TLS is being used to provide encryption, POP3S and IMAPS (note the addition of the “S” to denote the secure version of this protocol) operate on TCP ports 995 and 993, respectively.

To configure Exchange 2000 Server to use SSL for POP3 and IMAP, the SSL certificate must be imported into the Exchange server’s private certificate store. The Certificates MMC snap-in can be used to import the certificate, given the passphrase of the PFX file. Once the certificate has been imported, the POP3 or IMAP virtual server in Exchange System Manager must be configured to use the imported certificate and require SSL/TLS. The figure below, Figure 1, shows a screenshot of the IMAP virtual server configured for SSL/TLS.

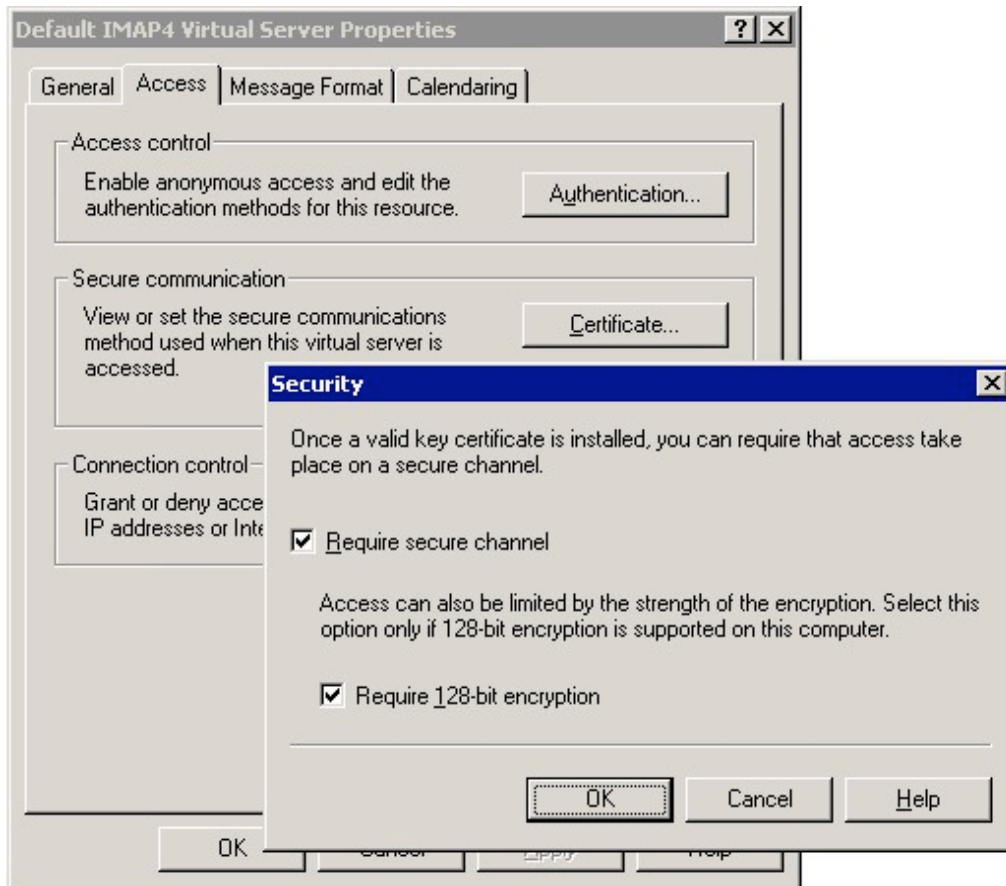


Figure 1. Configuring IMAP for SSL/TLS encryption

Because Stunnel will be used to provide the SSL/TLS functionality for SMTP, it is not necessary to configure the SMTP virtual server in Exchange System Manager.

Preparing Certificates for Use with Stunnel

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 move certificates from a Windows-based server running Exchange 2000 Server to another server running Stunnel, the certificates must be converted.

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. These steps require the use of OpenSSL, an open-source SSL toolkit. (Note that these steps were confirmed using OpenSSL on both Red Hat Linux and Mac OS X; the steps should be very similar on Windows as well.)

1. If the certificate is not already available in PFX format, use the MMC Certificates 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) are 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, 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.

Configuring Stunnel

Stunnel can be configured to run manually (i.e., must be invoked by a user) or automatically (as a system service). This technical note describes how to invoke Stunnel for SMTPS functionality manually on a Red Hat Linux-based system. For Stunnel to run automatically on a Linux-based system when the system boots, inetd (or xinetd) or a system startup script would have to be

written. On the Windows platform, version 4.04 of Stunnel (the version discussed in this document) supports running as a Windows service; see the man page for more information and the appropriate command line switches.

The following Stunnel configuration file can be used to provide SMTPS functionality:

```
cert = /etc/stunnel/stunnel.pem
pid = /var/run/stunnel.pid
setuid = root
setgid = root
CApath = /etc/Stunnel
[smtps]
accept = smtps.company.com:465
connect = mail.company.com:25
```

This configuration file configures Stunnel with the appropriate certificate and certificate authority path, and instructs Stunnel to provide SMTPS functionality and forward the requests to a separate server, mail.company.com, on the standard SMTP port (TCP port 25).

Note that this Stunnel configuration file has Stunnel running as root. This is required because Stunnel needs to bind to a privileged port below 1024. To limit the potential security vulnerabilities that may be created by running a process as root, Stunnel can be configured to run in a chroot jail with the following configuration file:

```
cert = /etc/stunnel/stunnel.pem
chroot = /etc/stunnel/
pid = /stunnel.pid
setuid = root
setgid = root
CApath = /
[smtps]
accept = smtps.company.com:465
connect = mail.company.com:25
```

In this configuration file, the chroot option instructs Stunnel to run inside a chroot jail—this means that Stunnel cannot “see” or access anything outside of /etc/stunnel. Because the pid and CApath options are relative to the chroot jail, their settings are modified. Note that the cert option is not relative to the chroot jail.

Summary

When used together, Exchange 2000 Server and Stunnel can provide a complete solution for Internet e-mail clients using secure versions of POP3, IMAP, and SMTP. Exchange 2000 Server can provide the SSL functionality for POP3 and IMAP natively, while Stunnel can supplement Exchange’s built-in support to provide SMTPS support on a separate TCP port (which Exchange 2000 Server cannot natively do). This added functionality might be necessary in a variety of instances.

Other Notes

None

Related Articles/Resources

The following Microsoft Knowledge Base article contains more information about support for SMTPS in Exchange 2000 Server:

278339: XGEN: TCP/UDP Ports Used by Exchange 2000 Server

More information on Stunnel can be found at <http://www.stunnel.org>; more information on the OpenSSL toolkit 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.