



Mercurion Systems, Inc.

Information Technology Consulting and Support Services

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

SECURING RDP TRAFFIC WITH SSL

Products: Stunnel 4.04; Windows 2000 Server or Windows Server 2003; Mac OS X 10.2

Overview

Microsoft's RDP protocol, used by Terminal Services (or Remote Desktop on Windows Server 2003), has some built-in encryption functionality. In this regard, it is similar to other thin-client protocols, such as ICA (Independent Computing Architecture, created and maintained by Citrix Systems for use in the MetaFrame family of products). However, unlike ICA, Microsoft does not provide any mechanism for applying additional security protocols to RDP traffic (ICA provides native support for SSL [Secure Sockets Layer] encapsulation).

It is possible, however, to add SSL encryption and encapsulation to RDP traffic through the use of Stunnel, an open source SSL wrapper. Stunnel's support for the Windows platform means that Stunnel can be deployed directly on a server providing RDP-based services without the need for a UNIX-based system to terminate the SSL tunnel.

This technical document provides information on how to use Stunnel to provide SSL encryption and encapsulation for RDP traffic.

More Information

Several pieces of software were used in this scenario. These software packages included:

- Terminal Services (in Windows 2000 Server) or Remote Desktop (in Windows Server 2003)
- Stunnel 4.04 (available from <http://www.stunnel.org>) and OpenSSL 0.9.7c (available from <http://www.openssl.org>)
- SSL Enabler 1.0 (provides a Cocoa GUI front-end to Stunnel on Mac OS X 10.2)
- The appropriate RDP client (Remote Desktop Connection in Windows XP, Terminal Services Client in Windows 2000, etc.)

The sections below provide more information on how to configure these software packages together to encapsulate RDP traffic inside SSL.

Configuring the RDP Server (Terminal Services/Remote Desktop)

No configuration is *required* on the RDP server (the server running Terminal Services or Remote Desktop). However, to keep the experience as seamless as possible and to require the use of SSL

on the RDP connections, the TCP port for RDP connections should be modified to a port other than the default port (TCP port 3389).

To modify the listening port for RDP connections, use the following steps.

1. Using Registry Editor (regedt32.exe), navigate to the following key (the lines below have been wrapped for readability):

```
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Terminal
Server\WinStations\RDP-Tcp
```

2. Change the PortNumber value (default is 00000D3D, hex for 3389). Change this value in hex to the desired port number.
3. Restart the server or delete and recreate all RDP-Tcp connections for this change to take effect.

With the RDP connection listeners ready on a port other than TCP port 3389, Stunnel can now be configured and installed to listen on this port and accept all inbound RDP connections. By allowing Stunnel to listen on TCP port 3389, we avoid any potential configuration issues with RDP clients connecting to alternate ports.

Configuring Stunnel

Stunnel is fully supported on Windows 2000 Server and Windows Server 2003, and can be configured to run as a service. Use the steps below to install and configure Stunnel to protect inbound RDP connections with SSL encryption.

1. Download the Win32 binary for Stunnel 4.04 (a single executable file at the time of this writing) and copy it into the %SystemRoot%\System32 directory.
2. Download the Win32 OpenSSL binaries for the latest version of OpenSSL (a single executable file and two supporting DLL files at the time of this writing) and copy them into the %SystemRoot%\System32 directory.
3. Obtain the appropriate certificates (in PEM format) and copy them to the %SystemRoot%\System32\Stunnel directory (this directory will need to be created).
4. Create the following stunnel.conf configuration file (this example configuration file assumes that Windows was installed on drive C: into the Windows directory and that the RDP-Tcp connection listeners were reconfigured to accept connections on TCP port 3390):

```
cert = c:\windows\system32\stunnel\stunnel.pem
CApath = c:\windows\system32\Stunnel\.0
[rdp]
accept = 3389
connect = 3390
```

Note that the "CApath" line above must be modified to show the appropriate file name for the hash of the certificate for the certificate authority that issued the certificate referenced on the "cert" line. This hash can be obtained using the OpenSSL toolkit.

Be sure to save this stunnel.conf configuration file in the same folder as the stunnel executable itself (in this example, %SystemRoot%\System32).

5. Install Stunnel as a service using the command line “stunnel –install”.
6. Start the Stunnel service using the Services MMC snap-in.

Once the Stunnel service has started, you can verify its operation with the “netstat –a” command at a command prompt. The results of that command should show listening connections at both TCP port 3389 (Stunnel) and TCP port 3390 (the RDP connection listeners).

Configuring the RDP Client

As with the RDP server, no real configuration is required on the client. This is primarily due to the fact that Stunnel handles all the SSL-related work—the client needs absolutely no knowledge of SSL whatsoever. Also, the fact that the RDP server has been reconfigured to use a different port and Stunnel is listening on TCP port 3389 also assists in full compatibility and interoperability.

The paragraphs below discuss the use of SSL Enabler 1.0, a utility designed for Mac OS X as a graphical front-end to Stunnel. Using SSL Enabler, the client-side Stunnel configuration is greatly simplified. A screenshot of the main SSL Enabler window is found below in Figure 1.

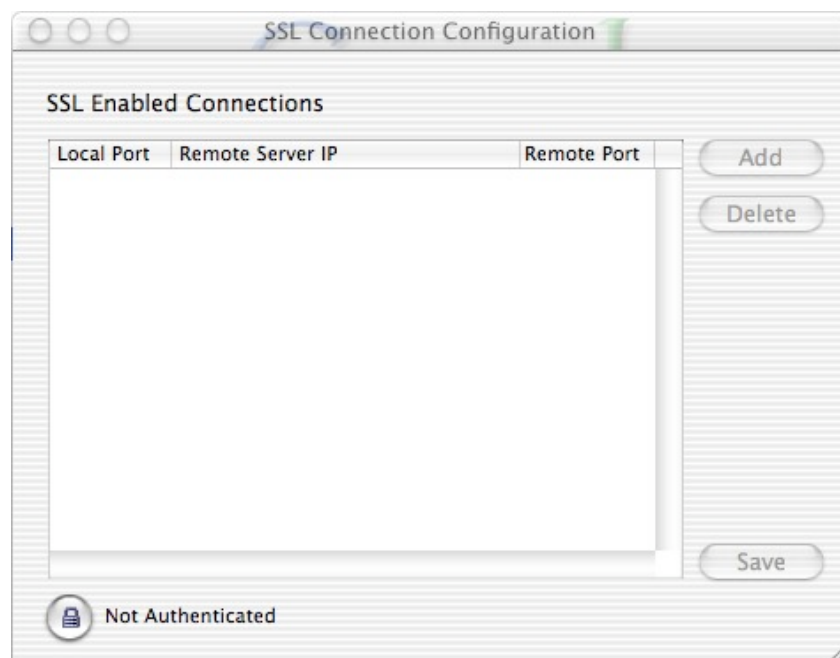


Figure 1. The SSL Enabler window

To configure the client for an SSL-wrapped RDP connection, first authenticate (using the lock in the lower left corner of the SSL Enabler window—this will require an administrative username and password). After this step is complete, the “Add,” “Delete,” and “Save” buttons will be enabled. Use the “Add” button to add the following information:

- *Local Port*: Use 3389. This prevents any potential problems with RDP clients that won’t connect on the standard TCP port 3389.

- *Remote Server IP:* Enter the fully-qualified domain name or IP address of the RDP server that is running Stunnel.
- *Remote Port:* Enter 3389 (or whatever port Stunnel was configured to listen on).

Once this information is entered, use the “Save” button to save this configuration and start Stunnel in the background. (Use the “ps -ax | grep stunnel” command in a Terminal window to verify the presence of an Stunnel process, and use “netstat -ta” to show listening connections on the local machine, which should include a listener on TCP port 3389).

Once Stunnel has been configured via SSL Enabler, launch the RDP client and connect to “localhost” or “127.0.0.1”. Stunnel will accept this connection (a connection being made to the Local Port specified above), encrypt/encapsulate it in SSL, then forward it to the specified port on the specified server (the Remote Port and Remote Server IP from above, respectively).

Other Notes

Multiple entries in SSL Enabler on different local ports would be required to service multiple remote RDP servers. Although Microsoft claims that the Mac OS X Remote Desktop Connection Client does not support connections to servers on an alternate port (other than TCP port 3389), real-world experience by Mercurion Systems suggests otherwise.

Related Articles/Resources

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>.

The following technical documents also provide additional information about Stunnel, SSL encryption, or other related technologies:

- Enhancing Exchange 2000 Server with Stunnel
- Securing ICA Sessions with SSL

The following Microsoft Knowledge Base article provides additional information on changing the TCP port for inbound RDP connections:

187623: How to Change Terminal Server’s Listening Port

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.