



Mercurion Systems, Inc.

Information Technology Consulting and Support Services

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

APPLICATION-SPECIFIC VPNS WITH STUNNEL

Products: Stunnel 4.x, Stunnel 3.x (as bundled with Mac OS X 10.2), SSL Enabler

Overview

Stunnel is an open source SSL wrapper that is designed to add SSL functionality to existing applications without requiring any modifications to the applications. Stunnel can be used in a variety of ways in order to add transport-layer security to existing protocols. Stunnel can be used on the server side to add SSL functionality to a service (for example, an e-mail service such as POP3 or SMTP) when existing clients already support SSL. Stunnel can also be used on both the client side as well as the server side, to encapsulate data in an SSL tunnel. By providing the ability to selectively encapsulate and encrypt traffic inside an SSL tunnel, Stunnel enables the creation of application-specific VPNs—that is, VPNs that are limited to a specific application or set of applications. This allows organizations to securely extend access to specific applications or resources to remote users for only those applications or resources they need.

This technical document describes how to use Stunnel to create application-specific VPNs.

More Information

Much of the information surround the use of Stunnel to create application-specific VPNs is similar to using SSH to create application-specific VPNs. For more information on the use of SSH to create application-specific VPNs, please see the technical document titled “Application-Specific VPNs with SSH.”

While the actual ports, servers, and such for using Stunnel to create SSL-based application-specific VPNs will vary based on the particular application or protocols involved, there are a few things that are common to all scenarios.

- Every tunnel runs between a TCP port on the local host and a TCP port on the server running Stunnel. The local host must have full connectivity with the server running Stunnel for the specified TCP ports.
- The server running Stunnel must have access to the remote server to which tunneled traffic will be directed. Preferably, this access should occur only over trusted subnets, since forwarding the traffic over insecure subnets would defeat the very purpose of using Stunnel. (Note that Stunnel can be configured to direct traffic to a TCP port on a different host or to a different TCP port on the same host.)

Stunnel must be properly configured on the server as well as on the client. The sections below describe how to configure Stunnel on the server and on a client. See the technical notes

referenced in the “Related Articles/Resources” section for additional information on configuring Stunnel in various scenarios.

Configuring Stunnel on a Windows-Based System

There is no graphical user interface (GUI) for configuring Stunnel; all configuration must be done with the `stunnel.conf` configuration file. A sample `stunnel.conf` file is found below; this file listens on TCP port 1494 and forwards traffic to TCP port 3389 on the same system. (Note that this would be a sample `stunnel.conf` file for a server-side configuration.)

```
CPath = c:\windows\system32\stunnel
cert = c:\windows\system32\stunnel\stunnel.pem
client = no
service = SSLTunnel
[rdp]
accept = 1494
connect = 3389
```

Once Stunnel has been configured, running “`stunnel -install`” will install it as a system service; this service can then be stopped and started through the Services MMC console.

Configuring Stunnel on a Mac OS X-Based System

Mac OS X 10.2 (“Jaguar”) includes Stunnel 3.x (as opposed to Stunnel 4.x), so the configuration is handled a bit differently. However, there is a GUI-based utility for configuring Stunnel on Mac OS X called SSL Enabler. Figure 1, below, is a screenshot of SSL Enabler running.

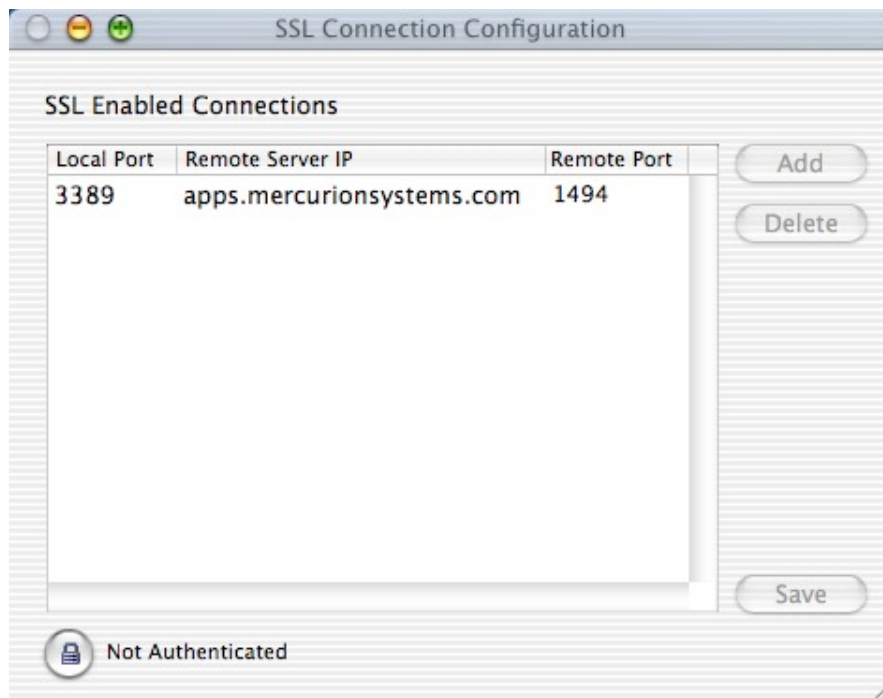


Figure 1. SSL Enabler on Mac OS X for configuring Stunnel

As shown in Figure 1 on the previous page, SSL Enabler allows for the creation of “SSL tunnels” or “SSL enabled connections.” Each SSL tunnel, much like an SSH tunnel, creates a local listening port. Connections to this local port (TCP port 3389, in this example) are wrapped in SSL and forwarded to the specified port on the remote host. The remote host must have an SSL-enabled service running on the specified port, or Stunnel must be running to terminate the SSL tunnel and forward the unencrypted traffic to the appropriate service.

Other Notes

None

Related Articles/Resources

A specific example of using Stunnel to create an SSL-based application-specific VPN is described in the technical document titled “Securing RDP Traffic With SSL.”

The technical document titled “Enhancing Exchange 2000 Server With Stunnel” also describes the use of Stunnel to add and/or modify SSL functionality for Exchange 2000 Server.

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.