



# 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 SSH

*Products: Various SSH servers & clients*

### Overview

Secure Shell (SSH) is a well-known and widely accepted way of providing secure, encrypted logons to UNIX and UNIX-related derivatives (such as Linux, OpenBSD, and Mac OS X). In addition to providing secure terminal services, the SSH protocol suite also typically provides secure FTP (SFTP) functionality for secure file transfers between systems. SSH also offers the ability to create application-specific virtual private networks (VPNs) by encapsulating specific ports or protocols inside the encrypted SSH session. By using this ability to create encrypted tunnels with SSH, organizations can securely provide e-mail, web, and other services to remote users.

This technical note describes (in general terms given the wide range of SSH servers and SSH clients) how to create application-specific VPNs using SSH tunneling.

### More Information

Version 2 of the SSH protocol provides for the ability to create SSH tunnels, i.e., to pass other TCP-based protocols inside an encrypted SSH session. As long as the other protocols run on standard ports and have predictable traffic patterns, it is generally possible to encapsulate such protocols inside SSH for secure delivery to a remote host. (Note that SSH version 1 is considered less secure than version 2 and does not provide support for SSH tunnels.)

While the actual ports, servers, and such for forwarding TCP traffic inside an SSH tunnel varies from situation to situation, 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 a remote host. The remote host is generally not the SSH server, but a remote host to which the SSH server has direct and secure connectivity.
- Opening a local TCP port less than 1024 on a UNIX, Linux, or derivative system (such as Mac OS X) requires root access. For this reason, the local port is often not the same as the remote port. (Refer to the examples below.)
- The SSH server 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 SSH tunnels in the first place.

Because of the wide variety of SSH clients and servers running on a wide variety of platforms, it is difficult, if not impossible, to specifically list how to establish an SSH tunnel. However, a few key examples can be found below.

## Providing Secure E-Mail Services

The use of SMTP and POP3 for sending and receiving Internet e-mail, respectively, is very common. However, both protocols lack the ability (without the use of SSL) to protect sensitive information in the protocol stream. For example, POP3 passes the username and password in clear text between the POP3 client and the POP3 server. Similarly, messages passed to an SMTP server are also in clear text.

The use of an SSH tunnel can alleviate those security concerns. The diagram below, Figure 1, illustrates how this might work in such a scenario.

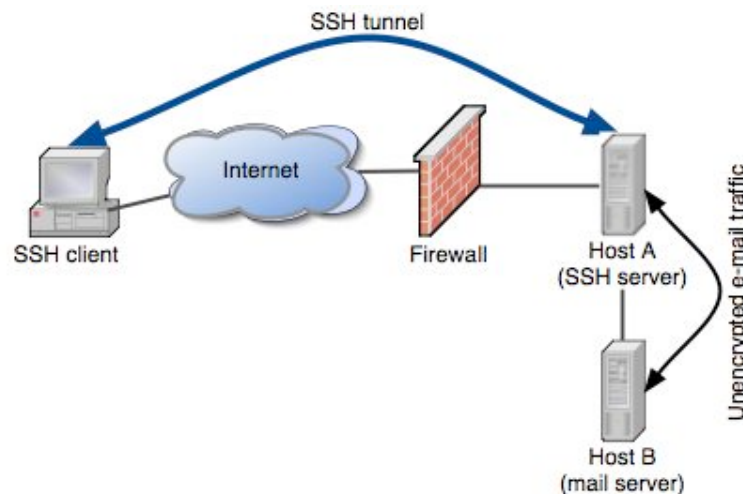


Figure 1. Using an SSH tunnel to secure e-mail traffic

As shown in Figure 1, Host A is the host running SSH. In addition, note that Host A has direct connectivity to the SMTP/POP3 server over a trusted subnet. (Note that unless SSL is employed, the POP3 and SMTP traffic between the SSH server and the e-mail server is not protected by encryption and therefore should only cross trusted subnets. This was pointed out earlier.) The client is running an SSH client that supports SSH version 2 and SSH tunnels.

The client establishes an SSH session to Host A, and along with the session creates a tunnel that forwards traffic received on TCP port 2525 of the local host to TCP port 25 on Host B (the remote SMTP/POP3 server). Likewise, the tunnel also forwards traffic bound for port 110 of the local host to TCP port 110 on Host B. (The selection of high-numbered ports avoids the need for root access, as mentioned earlier.) Once the session has been established, the user can reconfigure his or her e-mail client to use TCP ports 2525 and 110 on the local system to send and receive e-mail, respectively. When the local host receives a request on TCP port 2525, SSH picks up that traffic and forwards it inside the encrypted SSH session to Host A (the remote SSH server). Host A then decrypts the traffic and passes the traffic to Host B.

This behavior is exactly like any other VPN protocol, including IPsec, PPTP, and L2TP over IPsec, except that SSH is only encrypting specific protocols (such as SMTP or POP3) instead of all traffic.

On a Mac OS X 10.2 system (which is based on a derivative of FreeBSD), the command to create an SSH session and establish an SSH tunnel would look something like this:

```
ssh -l username -c cipher-type -p port -2 -L 2525:remotehost:25  
-L 1110:remotehost:110 -N -f ssh-server
```

The command would look very, very similar (if not exactly the same) for a system running Linux or other UNIX-based OS.

Note that some of the command-line parameters could be omitted (for example, the `-l` [username] option, the `-p` [port] option, the `-c` [cipher type] option, and the `-2` [protocol version] parameter) since the defaults will generally work just fine.

From a Windows-based system, an application such as SSH Communications' SSH Client or TeraTerm Pro are needed to establish the SSH session and create the SSH tunnels. The screenshot below, Figure 2, shows the SSH Communications SSH client being configured for a tunnel to forward SMTP traffic (TCP port 2525 on the client being forwarded to TCP port 25 on the remote server).

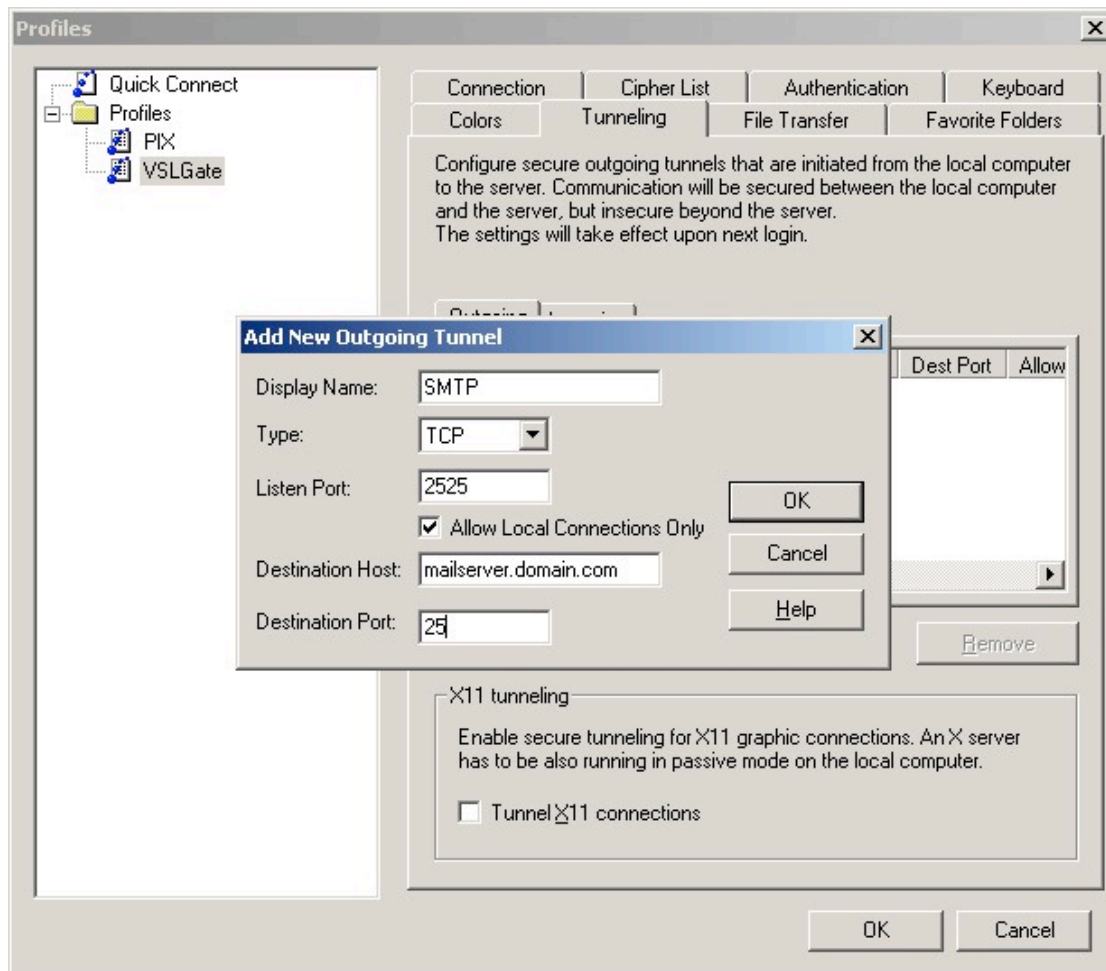


Figure 2. Configuring a Windows SSH client for an SSH tunnel

## Secure Citrix MetaFrame Access

Using SSH tunnels, it is possible to encapsulate (and thus protect with encryption) Citrix ICA traffic between an ICA client and a Citrix MetaFrame server. For example, the following command on a Mac OS X or Linux client would forward Citrix ICA traffic (on TCP port 1494) to a remote server:

```
ssh -l username -c cipher-type -p port -2 -L 1494:remotehost:1494  
-N -f ssh-server
```

ICA is not an inherently insecure application; quite the opposite, actually, since Citrix offers proprietary SecureICA encryption (up to 128-bit RC4) as well as SSL/TLS encapsulation (again, with encryption up to 128 bits). For more information on using SSL/TLS to secure Citrix ICA sessions, refer to the technical note titled, "Securing ICA Sessions with SSL."

## Secure WebDAV File Collaboration

WebDAV (Web Distributed Authoring and Versioning) is a useful way to provide file sharing/collaboration functionality across platforms that support WebDAV. However, while a variety of platforms support WebDAV (including Windows XP Professional and Mac OS X), not all platforms support some form of encryption to protect the WebDAV traffic.

Using an SSH tunnel, traffic directed to port 8080 on the local host will forward to TCP port 80 on a remote host (for HTTP, since WebDAV is an extension of standard HTTP) encapsulated in an encrypted SSH protocol stream and therefore protected during transit from the client to the WebDAV server. This command, executed on a Mac OS X or Linux machine, would establish just such an SSH tunnel:

```
ssh -l username -c cipher-type -p port -2 -L 8080:remotehost:80  
-N -f ssh-server
```

Once this tunnel was established, users on the client system could reference *http://localhost:8080* and be connected to the WebDAV server through the encrypted SSH tunnel.

## Other Notes

On UNIX and UNIX-derived systems (such as Linux and Mac OS X), the commands listed in this technical note would cause the ssh process to "fork" into the background. To end these SSH tunnels, use the "ps" command to find the process ID, then use the "kill" command to end that process ID.

## Related Articles/Resources

More information on using SSL/TLS to add security to Citrix ICA sessions can be found in the technical note titled "Securing ICA Sessions with SSL."

## 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.