Secure Transfers

Securing file transfers using encrypted connections across the Internet.

When discussing secure file transfers using Rumpus, the difference between FTP and WFM (Web File Manager) transfers is crucial. Thanks to the need for secure transactions in e-commerce, and the fact that HTTP (the Web) is a simpler protocol from a networking perspective, secure file transfers via the Rumpus WFM are much easier to implement and are consistently and almost universally supported by modern Web browsers.

Unfortunately, there is no consistent and widely implemented encryption standard used by FTP clients. Options for encrypted FTP exist, but they can be complicated to implement and will restrict the choice of FTP client software that can be used. The most commonly used encryption methods used by FTP clients are SFTP and FTPS, and both are described later in this article. But since Web browsers will provide the best secure solution for most clients, we'll begin there.

WFM (Web-Based) Encrypted File Transfers

All mainstream, modern Web browsers support trusted and encrypted connections via SSL (Secure Sockets Layer). Rumpus does not include native SSL support, but does include special features that enable it to work with SSL "tunnels", separate applications that accept and process SSL connections for servers like Rumpus. One such SSL tunneling application, "Stunnel" can be easily installed and configured using Maxum's "iAssist" product. The rest of this article will assume you are using iAssist to install Stunnel, though most of the information that follows is applicable to any SSL tunneling software you choose.

SSL Certificates

SSL provides not only encryption of file transfers, but server (and client) authenticity as well. In other words, not only will your clients be assured that their data is transferred using an encrypted connection, but that the file is being sent to the correct server that is owned by a reputable organization. This requires that you obtain a "certificate", which is a digital file that describes your organization and server and is "signed" by a trusted authority. You can pay for an authority to provide you with a trusted certificate, or you can "self-sign" the certificate yourself. Certificates signed by a known authority will usually be automatically trusted by common Web browsers, while self-signed certificates will cause browsers to display a warning message to users declaring that they are connecting to a non-trusted server.

For complete details on generating and purchasing SSL certificates, see the "SSL:Certificates" article in iAssist. Even if you choose not to purchase iAssist, download the demo copy from the Maxum Web site and review the SSL articles included. The SSL:Certificates article includes step by step instructions for generating a "certificate request" (a file you can send to a signing authority to obtain a trusted certificate) and for creating self-signed certificates.

Installing And Configuring The Tunneling Software

iAssist, or whatever SSL tunneling software you choose, also includes details on installing and configuring the tunnel. The tunnel should be set up on the same computer running Rumpus. You will need to install your certificate (again, according to the instructions in iAssist) and configure a single tunnel for port 443 (the standard HTTPS port). The tunnel should be configured to accept SSL connections on port 443 and direct the connection to the defined WFM port as specified by the "Port Number" field of the "Web Server" tab on the "Web Settings" window in Rumpus (probably 80 or 8000).

It is important to note that the SSL tunnel must be run on the Rumpus server itself and the tunnel must be set up for use on the standard HTTPS port 443. This will not only make accessing the server via SSL easier for your users (by making the access URL as simple as possible) but will allow Rumpus to correctly specify URLs in WFM-created pages.

Rumpus Settings

To have Rumpus handle tunneled connections automatically, open the "Web Settings" window, switch to the "WFM Options" tab, and check the "Support Tunneled SSL Connections" checkbox. When this option is checked Rumpus will assume that connections made from the server itself (specifically, from the tunneling software) should be handled as SSL connections. This causes a couple of changes in WFM behavior for these connections, the most important of which is to create "https" URLs in WFM pages.

After checking this option, connecting to the Web File Manager using a browser on the server itself will always result in an SSL-encrypted session. This is usually not a problem in real world use (there is usually no reason to use the WFM on the server machine itself), but it is an important point to keep in mind for testing. After enabling the "Support Tunneled Connections" option, be sure to test the Web File Manager from another computer on your LAN, and not the server itself.

Accessing The Secure WFM

With the tunneling software installed, your certificate in place, the tunnel configured, and Rumpus set to allow tunneled connections, the server is ready. All normal Rumpus WFM options will work exactly as they do using non-encrypted connections, and users can connect securely by entering a URL that begins "https://", rather than "http://" (or leaving the protocol specifier off). If your normal WFM access uses a nonstandard port (any port other than 80), be sure to leave the port number off, since the SSL connection will always be performed on the standard https port. So, to connect to your Rumpus WFM, you would use a URL of the form:

```
https://files.yourserver.com/
```

Of course, replace "files.yourserver.com" with the IP address or domain name of your Rumpus server.

That's it. Once connected, browsers will show the "lock" icon to signify the secure connection, and all activity, including file uploads, downloads and directory listings, will be transferred securely.

SSL-Encrypted FTP (FTPS)

One option for securing FTP connections (including both the control and data connections of the FTP session) is SSL encryption. While Rumpus does not support SSL encrypted sessions directly, it is fully compatible with the 3rd party application "Secure FTP Wrapper", from GlubTech. This solution is easy to set up and test, and offers a high level of additional server security.

Installing Secure FTP Wrapper

To get started, you should have an installed and running Rumpus FTP server. Before attempting to apply the secure wrapper, make sure you can connect using a standard FTP client. In this case, we recommend that you use the Mac FTP client Transmit, from Panic software, as it supports both standard and SSL-encrypted FTP, allowing you to test both interfaces to your FTP server.

http://www.panic.com/transmit/

Next, download Secure FTP Wrapper from the GlubTech Web site.

http://www.glub.com/

The Secure FTP Wrapper download includes a package installer. Run the installer and read the instructions displayed. When using Secure FTP Wrapper in trial mode, you will receive a demonstration license via e-mail, which needs to be copied to the correct location on your boot volume. The easiest way to do this is to open the Secure FTP Wrapper folder in the Finder by opening the Terminal application (in the folder "/Applications/Utilities") and entering the command:

```
open /usr/local/ftpswrap
```

With the folder open in the Finder, simply drag the license attachment from the e-mail into the "ftpswrap" folder.

Once you have run the Secure FTP Wrapper installer and copied the demo license file, restart the server to start the Secure FTP Wrapper application.

When the server restarts, the SSL wrapper will be automatically started. Start Rumpus, if it isn't set to automatically start, then login using an SSL-capable FTP client such as Transmit or GlubTech's Secure FTP. In Transmit, for example, supply the address or domain name and your user account name and password normally, then select "FTP With Implicit SSL" from the "Protocol" pop-up menu.

SFTP

While it is a common and understandable misconception, SFTP does not provide a secure mechanism for communicating with an FTP server. It is, in fact, a separate protocol built into SSH (Secure SHell), a software component built into the Unix underpinnings of Mac OS X. So, while it is possible to tunnel FTP sessions in SSH-encrypted connections, this does not permit Rumpus to be used with SFTP-capable clients. In addition, setting up SSH tunnels is cumbersome and must be performed on both the server and client, further reducing the attractiveness of SSH encryption for FTP use.

Therefore, Rumpus does not currently support SFTP. For secure transfers, we strongly recommend HTTPS using the Web File Manager and/or SSL-encrypted FTP, as described above.