
SFTP – The Secure File Transfer Protocol

Introduction

SFTP, the *Secure File Transfer Protocol*, is quickly gaining momentum as the de-facto standard for transferring files securely over the Internet. Many companies and ISP's are unplugging their legacy FTP servers in lieu of this emerging secure protocol. This paper provides a technical overview of the protocol, and outlines the benefits of its use.

Overview

The *Secure File Transfer Protocol*, or SFTP, is not based on the industry standard FTP, *File Transfer Protocol*, as defined in RFC-959. It is a completely distinct file transfer specification developed by the Secure Shell Working Group and SSH Communications Security Corp. Built on the Secure Shell (SSH) Protocol, the purpose of the protocol is to provide the ability to have secure, efficient, file transfer occurring over an SSH encrypted pipe or tunnel. The general idea is to connect to a remote SSH/SFTP server on port 22, perform a secure SSH v2 handshake with the remote server, and then all future communications would take place through the existing encrypted tunnel. No new connections would need to be established, as they are when using FTP. As of the date of this writing, most SSH servers support either version 3 or version 4 of SFTP as its file transfer protocol.

No more data channels

A key differentiator between standard FTP and SFTP is that an SFTP conversation occurs entirely over a single secured channel. Where FTP uses both a *command channel* (usually occurring on port 21) to exchange commands and a separate *data channel* (occurring on an arbitrary negotiated port, or sometimes on port 20), SFTP has the ability to perform both *command* and *data* conversations over the same channel.

During an FTP session where data needs to be transferred between the client and server, or server and client, the two endpoints will usually agree on a new IP/Port combination that will be used to transfer the data. Once the IP/Port is agreed upon, one party will listen/wait on that IP/Port and the other party will attempt to establish a connection to the listening party. Once the connection has been established, data can then be transferred. After the data has been transferred, the data connection is closed and a status code is returned to the client via the control connection. There are many security issues with this process. When the data connection is established, the two parties have no assurance that there is not a man-in-the-middle attack occurring and that someone is hijacking the data. This is a disadvantage of using FTP.

During an SFTP session where data needs to be transferred between the client and server, the client will merely sends an OPEN command to the server along with the name of the file to be transferred. Once the file has been opened, the client will then send the data to the server (or the server will send the data to the client). After the file has been transferred, the client will send a CLOSE command to the server and the file transfer has been completed, all over the existing connection. Since the entire file transfer was completed over an existing secure channel, both client and server can be assured that no man-in-the-middle attack has occurred.

Firewall Integrity

During the IP/Port negotiation phase of a standard FTP session, the client and server need to take into account the presence of any firewalls that may exist between the client and server. If any firewalls are present, those firewalls would (in most cases) need to open up additional ports to allow for the FTP *data connection* traffic to flow. More holes in a firewall mean greater potential security vulnerabilities.

While it's true that RFC-959 defines port 20 as the standard port for *data connections*, very few FTP clients rely on the default port. Most will issue the PASV or PORT commands to the server to generate a new port to be used for the *data connection*. Some FTP Servers, such as [Titan FTP Server](#), provide functionality so that the range of ports available for *data connections* can be limited, which helps with firewall integrity, but it does not completely alleviate the problem.

SFTF bypasses this issue altogether by only using a single port, port 22, on the firewall. All command and data traffic flows over the same encrypted pipe, through just one open port, eliminating the need to have large holes in your corporate firewall. This reason alone is causing many companies to opt for an SFTF server instead of standard FTP.

What about FTPS, isn't that secure?

FTPS, or *FTP over SSL*, is also an alternative to standard RFC-959 FTP. While FTPS does overcome many of the security concerns inherit in standard FTP, it cannot bypass the need for multiple control and data connections. It can also cause further performance issues. Since both the control and data connections need to be secured, the client and server will need to perform an SSL handshake during the establishment of each data connection. This is a time consuming process and can adversely affect overall performance of the server and client.

Conclusion

SFTF is quickly becoming a proven solution for secure file transfer. By using an industry standard encryption protocol such as SSH, many client developers are able to incorporate SFTF support into their existing FTP client software. Also, with the ability to perform both command and data conversations over a single pipe, many network administrators and integrators are plugging the holes in their firewalls and pushing for SFTF.

About South River Technologies

South River Technologies, develops Internet File Management and Collaboration Solutions, has full SFTF support in its products. [WebDrive](#), the world's first drive mapping client based on Internet protocols, supports SFTF. Also, [Titan FTP Server](#), SRT's enterprise class secure server product, is the first FTP server for Windows that supports SFTF version 3 and version 4.

For more information about South River Technologies and our products, please visit us on the web at <http://www.southrivertech.com/>.

The information contained in this document represents the current view of South River Technologies on the issues discussed, as of the date of publication. This White Paper is for information purposes only. SRT makes no warranties, express or implied, as to the information in this document. This White Paper may not be reproduced in any form without written permission from SRT.

Titan FTP Server and WebDrive are trademarks of South River Technologies, Inc. All other trademarks are the property of their respective owners