HOME SHOPPING NETWORK

B2B Supplier Data Exchange

Secure FTP Specification

Data Exchange partners

VERSION 2 - 5/31/05

HSN FTP for Supplier Data Exchange

HSN uses FTP for our Data Exchange partners to reduce VAN usage and charges. The first connection type is secure encrypted FTP with SSL certificates. Also known as FTPS.

HSN also provides an SFTP server based on OPEN SSH protocols, with SSH2 (Secure Shell) with AES encryption over a single connection to encrypt file transfers. The SSH2 layer guarantees encryption and protection for this protocol. This type of connection is high overhead and slower data transfer. See Section 5, page 7 for details on SFTP

FTPS with SSL Certificates

FTPS Clients

http://www.glub.com/products/secureftp/ has a multiplatform FTPS client including a scriptable Command line interface. The Glub-Tech source code is the basis for the Secure FTP server (SSL) that HSN uses.

http://www.ipswitch.com/Products/WS_FTP/ has a common Windows-only GUI client.

FTPS Configuration

- HSN’s test ftp server is: ftp://hsntestedi.hsn.net
- Connections from the client to the HSN FTPS are using:
- Control connection over port 990, Encrypted data connections on ports 3000-3200
- PASV or Passive mode is required

FTPS Process

Once a Data Exchange partner is ready to test connectivity, HSN will supply a user login and password.

A user directory (same as the login) will be setup. All files will be pushed and pulled from this directory. After authentication is complete with the FTPS server, the client must change to this directory.

For testing, a test PO file will be placed in that directory. You should test pulling this file, uploading files of your own, and deleting files. Data Exchange partners will be expected to delete production files after they have picked them up.

Files uploaded by the client will be moved for translation as soon as upload is complete, if they follow the HSN FTPS naming convention. Files that are not properly names will be ignored and left for the user to retrieve on their next pull.

File names will start with a two letter identifier, contain a distinct name, and end with a 3 letter suffix. i.e. SH0401031254.xml

- PO - Purchase Orders
- FA – Functional Acknowledgements
- SH – Shipping Confirmations
- CD – Credit Debit memos
- RA - Remittance Advise

- .edi – EDI/X12 file format
- .csv – HSN Comma separated file format
- .xml - HSN XML file format

In production, expect to retrieve and process Orders, Acknowledgements, and Remittance Advise files.

Once the testing is complete, HSN can move the login to the production environment, which is just a server name change.

1. Why use FTPS

The FTP (File Transfer Protocol) protocol is one of the most popular, and easy to implement methods of data transfer. With the addition of SSL certificates, a secure method for passing encrypted data over the internet is available without the use of Leased Lines or the need to exchange keys. This is also based on RFC standards and has been adopted by open-source software developers for low cost, quick implementation.

2. How the HSN FTPS Server Works

2.1. FTP

The File Transfer Protocol (FTP) has been around since the 1970's and was one of the first efforts to create a standard means of exchanging files over a TCP/IP network. The base specification is RFC 959 and is dated October 1985. There are some additional RFCs relating to the FTP protocol. You can find a good overview of all technical documents relating to the FTP protocol at the RFC Sourcebook.

2.2. The FTP Protocol Modes

The FTP protocol is a complex protocol because it uses a control connection (the primary connection) and a data connection (the secondary connection). How the data connection is made depends on the FTP protocol *mode*. There are two FTP modes:

- PORT or Active Mode
- Passive or PASV Mode

HSN’s FTPS server supports PASV mode only for easy setup for new trading partners.

2.3. The connection and ports

The control connection is the communication path between the USER and SERVER for the exchange of commands and replies. This connection follows the *Telnet Protocol*.

When an FTP client wants to exchange files with an FTP server, the FTP client must *first* set up the control connection. The client makes a TCP connection from a random unprivileged port N (N > 3000) to the FTP server's well known command port 21. **HSN’s FTPS server uses port 990 for this connection.**

It is important to note that the protocol requires the control connection to remain open while the data transfer is in progress. It is the responsibility of the user to request the closing of the control connection when finished using the FTP service. However, it is the server who takes the action to close the control connection.

The data connection is the communication path between the USER and SERVER for the exchange of the *real* data, being directory lists and files. The data connection is initiated from the client (passive mode).
The client sends a PASV command to the server. This command asks the server to "listen" on a data port (which is not its default data port 20) and to wait for a connection rather than to initiate one. The server will send a reply including the host (IP address) and port number (unprivileged port > 3000) this server is listening on. The client will then establish the data connection from that port to the IP address and port number learned from the PASV reply.

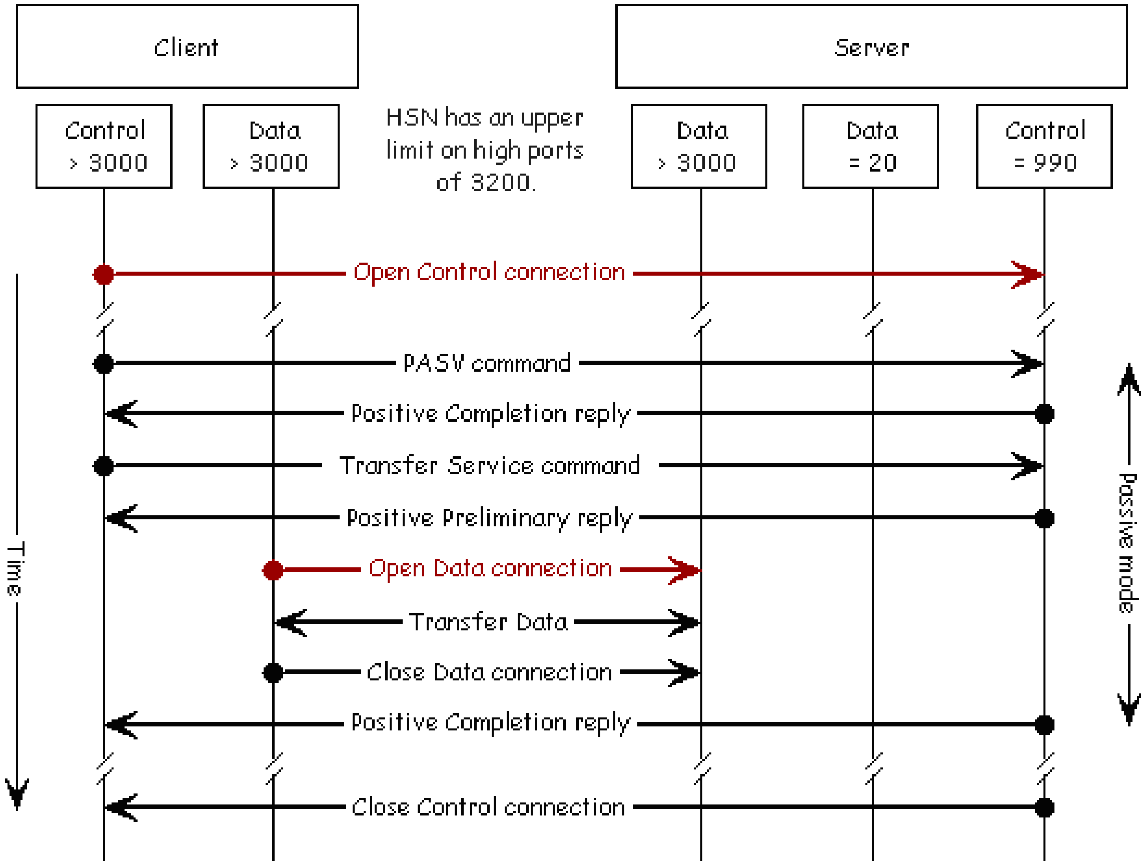
It is important to note that the data connection will only be established upon receipt of the reply to the Transfer Service commands such as LIST, RETR, STOR. So, the FTP mode must be sent by the client before sending the appropriate Transfer Service command. Also, those Transfer Service commands will get more than one reply: first a 'Positive Preliminary Reply' indicating the data transfer may start and a 'Positive Completion Reply' after ending the data transfer.

2.4. Passive mode

At the client side the port requirements aren’t that different from those required by the PORT mode FTP client, with the exception that the PASV mode FTP client requires outbound access to high-number TCP ports. This is a tremendous advantage for client security. The direction of the connection is outbound and is considered less hazardous with a good basic packet filter.

High-numbered ports 3000-3200 *must* be opened.

Schematically the information flow is as follows:



3. Secure FTP FTPS FTP-SSL

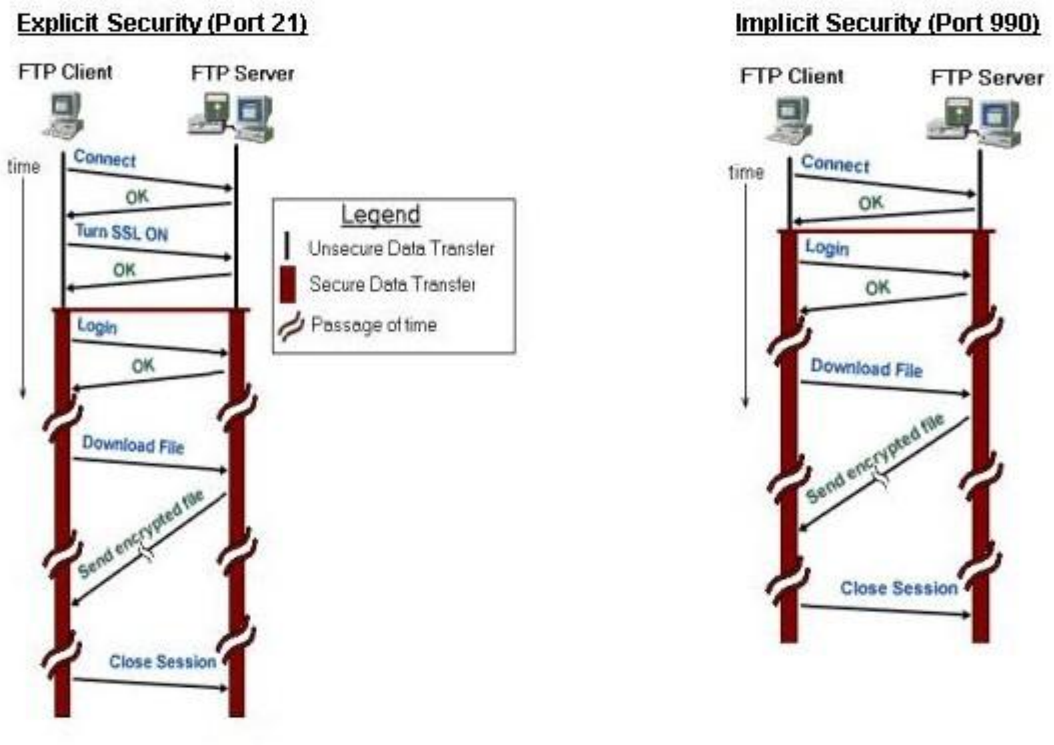
The document RFC2228 defines Security Extensions to the FTP protocol. These extensions provide strong authentication, integrity, and confidentiality on both the control and data channels with the introduction of new optional commands, replies, and file transfer encodings.

A widely implemented security extension to the FTP protocol is the use of the SSL (Secure Sockets Layer) version 3.0 or TLS (Transport Layer Security) version 1.0 protocol. HSN uses the SSL version, which runs above the transport layer (TCP/IP), but below the application layer protocol. Therefore, it is relatively easy to implement for securing protocols such as HTTP, Telnet, POP3, IMAP4, SMTP and FTP.

In FTPS client and server implementations, there exists at least two distinct mechanisms by which the SSL security extension is initiated: explicit (active) and implicit (passive) security:

- **Explicit Security:** In order to establish the SSL link, explicit security requires that the FTP client issues a specific command to the FTP server *after* establishing a connection. **HSN does not support this mode.**
- **Implicit Security:** Implicit security automatically begins with an SSL connection *as soon as* the FTP client connects to an FTP server. In implicit security, the FTP server defines a specific port for the client (TCP Port 990) to be used for secure connections.

HSN FTPS requires SSL Implicit. Consider implicit security as "always on" and explicit security as "turn on". The following diagram contrasts implicit and explicit SSL connections:



When using FTP over SSL, the control connection is encrypted and therefore unreadable for entities other than the FTP client and the FTP server. As a consequence, NAT/PAT devices and firewalls can no longer monitor the negotiation of the data connection.

4. Sample Connection Log

```
c:\ftps\> ftps
ftp> open hsntestedi.hsn.net
Attempting to make an implicit SSL connection to hsntestedi.hsn.net on port 990.
220-Glub Tech Secure FTP Wrapper (v. 2.5.4)
220 BridgeGate FTP Server.
PBSZ 0
200 PBSZ Command OK. Protection buffer size set to 0.
PROT P
200 PROT Command OK. Using Private data connection.
Name (hsntestedi.hsn.net:user): TESTUSER
USER TESTUSER
331 Password
Password:
Password: PASS **********
230 User TESTUSER logged in from hsntestedi.hsn.net
Type set to auto.
ftps> status
Securely connected to hsntestedi.hsn.net as TESTUSER;
Server's certificate:
  Issued by: VeriSign Trust Network, VeriSign, Inc.
  Issued to: HOME SHOPPING NETWORK, IT, HSNTESTEDI.HSN.NET
Data encryption: on;
Transfer mode: auto;
Verbose: on; Bell: on; Prompting: on; Globbing: on;
Hash mark printing: on; Connection type: passive;
ftps> ls
SYST
215 UNIX
TYPE A
200 type set.
PASV
227 Entering Passive Mode. (161,254,6,37,11,184)
LIST
150 Opening ASCII mode data connection.
226 Transfer complete.
TESTUSER/  .log         trash
ftps> cd TESTUSER
CWD TESTUSER
250 CWD command succesful.
ftps> ls
TYPE A
200 type set.
PASV
227 Entering Passive Mode. (161,254,6,37,11,185)
LIST
150 Opening ASCII mode data connection.
226 Transfer complete.
.log                      PO0302151728522.csv       tmp/
PO0211110237644.csv       PO0302153047459.csv
PO0226161015143.csv       PO0305154152853.csv
ftps> get PO0211110237644.csv
TYPE I
200 type set.
PASV
227 Entering Passive Mode. (161,254,6,37,11,186)
RETR PO0211110237644.csv
150 Binary mode connection for PO0211110237644.csv (21577 bytes)
[********************] 100% : 21.07 of 21.07 KB transferred
226 transfer complete
ftps> bye
QUIT
221 Service closing control connection.
c:\ftps\>
```

5. HSN SFTP or FTP + SSH server

SFTP (sometimes referred to as FTP+SSH) is a FTP security mechanism that uses SSH2 (Secure Shell) with AES encryption over a single connection to encrypt file transfers. The SSH2 layer guarantees encryption and protection for this protocol.

SFTP use the Triple DES (3DES) and Blowfish encryption algorithms. 3DES is a time proven cipher that provides strong encryption, while Blowfish is a *fast block* cipher that provides faster encryption. With SFTP, encryption is started before authentication -- therefore no passwords or other information is ever transmitted in the clear.

SFTP is similar to FTP in name only. It does not use the common FTP server processes (i.e., the "ftpd" or "wu-ftpd" Unix daemons) to establish a file transfer connection, but instead requires a "SSH2 with SFTP subsystem" on the server.

Using SSH and file transfers can result in up to 4x the byte traffic as using the FTPS implementation, this should be considered when choosing this type of data connection if your business is not on a dedicated connection for the Internet.

How does SSH work?

SSH works by the exchange and verification of information, using public and private keys, to identify hosts and users. It then provides encryption of subsequent communication, also by the use of public/private key cryptography.

In describing SSH, the term *client* means a workstation or PC that you are already logged in to, e.g., your own personal workstation or a group workstation that provides XDM session management for several X terminals. The term *server* means a secondary remote workstation that you wish to log in to, to do some work or transfer files: a login session server.

For more information, you can read the man page for SSH here:

http://www.openbsd.org/cgi-bin/man.cgi?query=ssh&sektion=1

SFTP Clients

Below are the certified SFTP Clients that may be used to connect to the SFTP Server:

- PSFTP (PuTTY command line interface for SFTP)
- WinSCP3 (http://winscp.net/eng/index.php)
- PrivateShell (http://www.privateshell.com)
- BitVise Tunnelier (http://www.bitvise.com/tunnelier.html)
- Pragma CL
- Other clients may work as well. If you find one that works that is not on this list, please let us know so we can add it.

SFTP Configuration

HSN's test ftp server is ftp://hsntestedi.hsn.net. Connect via SFTP to HSN using Port 24. use 'Password Authentication' mode from your SFTP client, not 'Private Key Authentication'. Enter the username/password given to you by HSN.

- HSN's test ftp server is: ftp://hsntestedi.hsn.net
- Control connection over port 24
- PASV or Passive mode is required

SFTP Process

The initial connection to each ftp server at HSN will prompt you to accept the server host key. The message may look similar to the following:

"The server's host key was not found in the cache. You have no guarantee that the server is the computer you think it is. The server's key fingerprint is: ssh-rsa 1024 c2:da:28:dd:b1:32:82:84:5c:6d:0f:4f:b2. If you trust this host, press Yes."

Your SFTP client may have the option to always accept, or just a yes option. Click yes or always to accept HSN's server host key.

Upon successful connection, you will see the directory listing for your account. Unlike normal FTP connections to HSN and FTPS (SSL) connections to HSN, you do not need to traverse into the directory used for your account name. You are automatically placed in your root directory. You may now send and receive files via this interface.