# Tumbleweed SecureTransport client does not send correct commands for SSL support

**THE INFORMATION IN THIS ARTICLE APPLIES TO:**

- EFT Server

**SYMPTOM**

When using SSL, users may not be able to connect to EFT Server using a Tumbleweed SecureTransport client. This occurs because the SecureTransport client fails to issue the appropriate FTP commands to secure the data channel connection that EFT Server expects.

For information about SecureTransport and EFT Server ***SFTP***, refer to : https://kb.globalscape.com/KnowledgebaseArticle11073.aspx

**RESOLUTION**

This happens because SecureTransport (ST) client violates a mandate of the RFC for FTP over SSL. The Tumbleweed client connects to the EFT Server, initiates SSL/TLS communication, and authenticates correctly with the EFT Server; however, any time the ST Client attempts to transfer a file or get a folder listing—that is, anything that requires a data channel connection—it fails to tell EFT Server to protect the communication of that data channel. At that point, EFT server is expecting plaintext data and does not attempt to perform an SSL Handshake on that data connection; however, the ST Client does attempt an SSL Handshake and of course that fails because the server does not anticipate it.

The RFC that specifies how SSL/TLS sits atop FTP is RFC 2228, found here http://www.ietf.org/rfc/rfc2228.txt Section 3, in the DATA CHANNEL PROTECTION LEVEL section (the "PROT" command), states that "The default protection level if no other level is specified is Clear."

It is the responsibility of the client software to clearly indicate to the server that the client wishes the Data Channel to be secured by issuing a "PROT P" command; further, that command requires a "PBSZ 0" command to be issued prior to the "PROT P". It is the sequence of these two commands together that turns ON the protection of the data channel, and this is the sequence that RFC compliant clients use to make this happen. Note that once this is set for a login session, all subsequent data channel connections remain protected. That is, you need only do this once, prior to your first transfer.

# Tumbleweed SecureTransport client does not send correct commands for SSL support

A typical sequence of client commands for a secure login and transfer session looks something like the following. This does not reflect the server responses. Missing Tumbleweed commands are indicated in RED:

```
AUTH SSL
```

```
USER foo
```

PASS bar

```
PBSZ 0
```

PROT P

```
PASV
```

STOR somefile.dat

Use a RFC 2228 compliant client such as CuteFTP Pro.

**WORKAROUND**

The current workaround for using the Tumbleweed SecureTransport client involves issuing the "PBSZ 0" and "PROT P" commands manually. To do this, type "quote" followed by the command. An example from the perspective of the SecureTransport client command line:

```
[ root@end-ipst03.bin ]# ./fdx -F off -V -p xsecureftp.xyz.com
```

220-Secure FTP.

Name (secureftp.xyz.co:root): AOR1468

331 Password required for AOR1468.

Password: XXXX

230 Login OK. Proceed.

Remote system type is UNIX.

Using 'binary' mode to transfer files.

fdx> quote PBSZ 0

200 PBSZ Command OK. Protection buffer size set to 0.

fdx> quote PROT P

200 PROT Command OK. Using Private data connection

fdx> ls

The lines in red must be manually inserted. If the client is using a script, the script file needs to be updated with those commands.

**Note:** A folder listing (ls or dir) is considered a data transfer, so the PBSZ and PROT commands must be issued prior to any folder listing.