Precipitation Processing System (PPS) Transition from FTP to FTPS

Executive Summary

Organizations will need to open/allow access to all ports in the range of 64000-65000 for the system 'jsimpsonftps.pps.eosdis.nasa.gov' and/or 'jsimpson.pps.eosdis.nasa.gov'

Currently the only server name that can be used with FTPS is: 'jsimpsonftps.pps.eosdis.nasa.gov' The other server name, 'jsimpson.pps.eosdis.nasa.gov', is still serving normal FTP connections, not FTPS. However, after the effective switchover date, both server names will serve only FTPS connections. All testing prior to this date should be done solely with the 'jsimpsonftps.pps.eosdis.nasa.gov'

A client program capable of FTPS interactions such as: 'lftp', 'wget', 'curl', or 'FileZilla' must be used to access the system as normal FTP clients will be unable to do so. Each of these clients has their own peculiarities of use, some simple examples of usage are provided in the full text below along with any known issues, version or operating system limitations. It is possible that certain programming or scripting languages such as Perl or Python might also have the necessary capabilities either natively or through additional libraries and/or modules, however no testing of that was done internally at PPS and no examples are provided.

The PPS implementation of FTPS on servers relies on what is called Explicit FTPS, rather than what is known as Implicit FTPS, and as such the initial connection uses port 21 – just like normal FTP does – rather than using port 990 as Implicit FTPS does.

Background

NASA information security management authorities decided that continued use of the File Transfer Protocol (FTP) should no longer be supported, even when used to provide access to publicly-available, non-sensitive information because login credentials are sent in clear text. For this reason they mandated that all FTP sites either convert to some form of encrypted login mechanism, or be shut down. PPS determined, after reviewing available options, that the option that provided the easiest mechanism to meet the NASA mandate, while hopefully allowing for the least amount of code changes needed by our user base was to transition from FTP to FTPS.

FTPS is basically FTP with the capability to provide encryption to either/both the login credentials and the data transfer. There are two implementations of FTPS known as Implicit and Explicit. The Implicit FTPS is the older, original form and is similar to how HTTPS differs from HTTP in that rather than using the existing FTP port (21) it works on a specific port (990) for connections and assumes that all traffic communicating with the port is encrypted from the very first connection. The newer method, Explicit FTPS, on the other hand uses the same port infrastructure as FTP and requires a STARTTLS command to be issued to begin the encryption channel. While it does not apply to PPS installations, this Explicit method allows for a server to support both FTP and FTPS on the same system concurrently. Current "best practices" in IT security indicate that Explicit FTPS is the preferred method of implementing FTPS, and as such, PPS has proceeded with this method in its implementation.

Encrypted Connected Lag Issues

Unfortunately, the imposition of encryption onto a FTP connection produces a noticeable lag. On interactive connections, the lag is usually only noticeable on the initial connection, while subsequent commands tend to be completed quickly. However, when operations are conducted as a series of independent connections, as is often the case with scripted retrievals, the lag applies to each connection individually. The internal tests conducted by PPS show that this lag time is usually in the 10-20 second range.

FTP(S) Data Port Issues

The existing, original PPS FTP servers have always used a limited range of ports (64000-65000) for data connections after initial connection has been made. This has never had any impact on users ability to connect because the negotiation sequence wherein the port is promulgated from server to client have been conducted in clear text. This clear text exchange of the next port to be used for data transfers allowed firewalls conducting stateful packet inspection (SPI) to be able to determine that a subsequent connection to port N was in fact related to a previously allowed connection and thus could proceed. Unfortunately, the requirement to encrypt the login exchange means that this information can no longer be gleaned by the firewalls because it is no longer visible to them. Therefore firewalls no longer see the initial connection and the follow-up connection to be 'related' and many, due to their own security settings, deny the data connection from proceeding, thus blocking access. In essence, this is one security system petulantly denying further access because another security system required encryption to be used, and now the first system isn't able to eavesdrop and figure out what's going on.

So, while PPS hasn't changed how the data ports work, or what sequence of ports is used, many users suddenly find themselves unable to access the server because their own firewalls are blocking the connections. The easiest, and most straight-forward, way to deal with this issue is for users to implement (or have their IT departments implement, depending on institutional size and requirements) a firewall rule that allows access to all ports in the range of 64000-65000 for the DNS names of: 'jsimpson.pps.eosdis.nasa.gov' and 'jsimpsonftps.pps.eosdis.nasa.gov' Doing this will ensure that connections can successfully work.

FTPS Client Examples

The following list is not meant to be exhaustive; these are simply the client programs that PPS used to test FTPS access to its server. The client programs have extensive option sets that PPS has not explored or provided documentation about, users are encouraged to investigate these options themselves as their need and desires determine.

lftp

The 'lftp' program was tested for manual FTPS connections, although the program does provide options for executing scripted connections, no example of this type is provided.

Example:

```
lftp jsimpsonftps.pps.eosdis.nasa.gov -u [user name],[password]
```

Notes: The 'lftp' program needs to be fairly recent. We found during investigations that the 'lftp' client in use on Redhat Enterprise Linux (RHEL) 6 systems (and its clones such as CentOS) did not have the required built-in security libraries necessary to interact with the FTPS server. However, RHEL 7 and RHEL 8 systems (and their clones) do not have this issue and were able to successfully connect.

curl

The 'curl' program was tested for directory listing and file download. All examples below are meant to be a single line, but due to line-wrapping in this document have been broken apart across multiple lines.

Example:

Directory Listing:

```
curl -4 –-ftp-ssl –-user [user name]:[password]
ftp://jsimpsonftps.pps.eosdis.nasa.gov/data/documentation/
```

File Retrieval:

```
curl -4 –-ftp-ssl –-user [user name]:[password]
ftp://jsimpsonftps.pps.eosdis.nasa.gov/data/documentation/nrtInstructions.pdf -o
nrtInstructions.pdf
```

Notes: The ‘curl’ program was the only one tested which was found to work across all three current versions of Redhat Enterprise Linux (RHEL) (and clone) systems, 6x, 7x, and 8x. Note that in the example above we have forced use of the Internet Protocol version 4 (IPV4) which may be necessary depending on how your system is provisioned. We have noticed that some systems have ‘default’ Internet Protocol Version 6 (IPV6) addresses set but which are not setup correctly and yet the system uses this IPV6 address regardless and thus return connection attempts are lost. The PPS FTPS server is setup to correctly handle IPV6 connections, but requires that IPV6 be setup correctly on the client end as well. If you’re sure that 1.) IPV6 is correctly setup on your system, or b.) IPV6 is not setup at all on your system, you may omit the ‘`-4`’ from the examples above.

wget

The ‘wget’ program was tested only for file download and not for directory listing, although that would be possible, it would require the resultant output to be ‘scraped’ to determine what the directory structures and files available were. The example below is meant to be a single line, but due to line-wrapping in this document has been broken apart across multiple lines.

Example:

```
wget -4 –-ftp-user=[user name] –-ftp-password=[password]
ftps://jsimpsonftps.pps.eosdis.nasa.gov/data/documentation/nrtInstructions.pdf
```

Notes: The ‘wget’ program could only successfully be used with Redhat Enterprise Linux (RHEL) (and clone) version 8, the versions found in version 7 and version 6 did not successfully work. Also note that unlike all other previous examples, ‘wget’ required the use of ‘`ftps://`’ instead of ‘`ftp://`’ in the URL provided to it. Also note that, as in the examples for ‘curl’ above, we have forced use of the Internet Protocol version 4 (IPV4) which may be necessary depending on how your system is provisioned. We have noticed that some systems have ‘default’ Internet Protocol Version 6 (IPV6) addresses set but which are not setup correctly and yet the system uses this IPV6 address regardless and thus return connection attempts are lost. The PPS FTPS server is setup to correctly handle IPV6 connections, but requires that IPV6 be setup correctly on the client end as well. If you’re sure that 1.) IPV6 is correctly setup on your system, or b.) IPV6 is not setup at all on your system, you may omit the ‘`-4`’ from the examples above.

FileZilla

We have successfully tested the graphical FileZilla client from both Windows10 and CentOS (RHEL clone) version 7. Currently there is no version of FileZilla available for RHEL version 8 and the version on RHEL version 6 does not have the updated security libraries necessary to connect.

Timeline

Currently normal FTP operations are being provided via the server: ‘jsimpson.pps.eosdis.nasa.gov’ and FTPS operations provided by the server: ‘jsimpsonftps.pps.eosdis.nasa.gov’ On date: 01 June 2020 all normal FTP operations will cease, and only FTPS connections will be allowed. At that time both server addresses, e.g. ‘jsimpson’ and ‘jsimpsonftps’ will provide FTPS access. At that time however, they will cease to be separate systems and will instead be collocated on a single host server. PPS will be maintaining the server name ‘jsimpsonftps.pps.eosdis.nasa.gov’ out of courtesy for those users who change over to FTPS earlier and do not wish to go back and modify their code to use the ‘jsimpson’ name again.