Using SFTP on the z/OS Platform

Thursday, December 10th 2009

Steve Goetze
Kirk Wolf

DovetailedTechnologies

http://dovetail.com
info@dovetail.com

Dovetailed Technologies

Our operating philosophy is to offer quality products licensed free of charge, along with world class support and consulting services.

- Co:Z Toolkit, which includes:
  - *Co:Z Launcher* – remote system cooperative processing
  - *Co:Z Dataset Pipes* – convert datasets to files
  - *Co:Z SFTP* – OpenSSH SFTP with z/OS exploitation
  - *Co:Z Batch* – full featured BPXBATCH replacement
  - *Co:Z FtpSshProxy* – tunnel ordinary FTP in SSH proxy
- T:Z Quickstart for Tomcat and z/OS
- JZOS - acquired by IBM in 2005 and now part of the z/OS Java SDK

Co:Z® UP TO A FRIENDLIER SOFTWARE PRICING MODEL:

FREE

Use the Co:Z Co-Processing Toolkit and our free support forum at no cost. Annual site support agreements start at just $4995.

DovetailedTechnologies
www.dovetail.com
+1.636.300.0901

Co:Z SFTP
OpenSSH secure file transfer with support for MVS datasets and SMF logging

Co:Z Launcher
Execute Unix or Windows processes from a batch job with access to MVS datasets

Co:Z Dataset Pipes
Flexible conversion of MVS datasets to/from pipes or Unix files

Co:Z FTP-SSH Proxy
FTP tunnelling over SSH

Co:Z Batch
A better BPXBATCH

Co:Z Co-Processing Toolkit
for z/OS | Dataset Pipes : SFTP : FTP-SSH Proxy : JZOS : DtlSpawn

Agenda

- What is SFTP and how it works with SSH
- How is SFTP different from FTP, and why use it
- Using IBM Ported Tools OpenSSH
  - Using z/OS as an SFTP server
  - How to use the SFTP client from a batch job
  - Enhancing Ported Tools OpenSSH with Co:Z SFTP
  - Transferring MVS data sets
  - Connecting with keys or passwords
    - Using SAF/RACF client certificates
  - Diagnosing problems and avoiding common pitfalls

What is SFTP?

- It's not FTP
- It's not FTPS (FTP with SSL/TLS)
- It's the Secure Shell (SSH2 specification) for file transfer
  - Most SSH implementations include an “sftp” command that has subcommands familiar to FTP users
  - The SFTP and FTPS wire protocols are **not** compatible

Terminology used in this presentation

- **SSH –** A draft internet standard defined by a group of related RFCs, aka “SSH-2”
- **SFTP** – SSH file transfer layer. SFTP implementations generally follow “draft-ietf-secsh-filexfer” version 3 or 4.
- **FTPS** – FTP with SSL/TLS; RFC-2228 et al.
- “**Ported Tools**” - IBM Ported Tools for z/OS OpenSSH; a non-chargeable, supported z/OS feature

Note: the old SSH protocol version 1 (“SSH-1”) has known security weaknesses, and should be avoided and disabled in your SSH servers if possible (the default in Ported Tools)

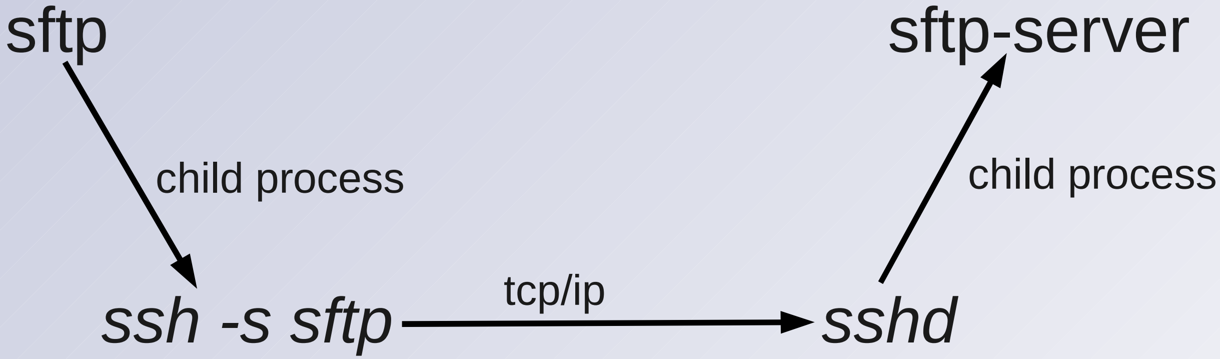
SSH features

- SSH provides:
  - A secure (encrypted) connection over one TCP/IP socket between a client and a server
  - The server's identity is authenticated using a public / private “host” keypair.
  - The client (user) can authenticate over the encrypted socket in one of several ways:
    - User public/private keypair
    - Password
    - GSS-API (Kerberos)
    - etc...
  - Data compression
  - Support for one or more simultaneous application “channels”

Types of SSH Application Channels

- “shell” (telnet)
  - not tn3270
  - a secure replacement for tty telnet (eg. PuTTY client)
- Remote command exec (redirect **stdin, stdout, stderr**)
- Port forwarding (and reverse forwarding, socks proxy etc)
- Subsystem: A named indirect command execution with binary **stdin**, **stdout** redirection:
  - *File transfer (sftp)*
    - A standardized ***packet*** protocol in the application channel
  - Additional subsystems can be configured

SFTP as an SSH Application/Subsystem



Copyright © 2009, Dovetailed Technologies

Implications

- The **sftp** command and **sftp-server** subsystem are not responsible for:
  - TCP/IP socket communications
  - Authentication: Key Exchange, Passwords, etc...
  - Encryption
  - Compression

SSH (+SFTP) Implementations

- **OpenSSH** – Free, open-source.
  Included on Unix/Linux distros; available on Windows.
  - IBM Ported Tools for z/OS includes a port.
  - Co:Z SFTP is a port of sftp and sftp-server for z/OS.
- **PuTTY –** Free, open-source Windows client.
  - WinSCP is a graphical Windows client that uses PuTTY
- **SSH Tectia** (SSH Communications) – Windows, Unix, Linux, z/OS
- **SecureCRT, SecureFX** (Van Dyke) - Windows, Unix, Linux
- ... (many others)

SSH Implementations are generally **very compatible**

Important differences between SFTP and FTPS

- **Host Authentication:**
  - FTPS - SSL/TLS (X.509 PKI server certificates)
  - SSH – Public/private DSA or RSA keypairs

- **User Authentication:**
  - FTPS - passwords, X.509 PKI client certificates, GSS-API
  - SSH – passwords, DSA or RSA keypairs, GSS-API, PAM, ...

- **Note:** IBM Ported Tools OpenSSH only supports a subset of user auth mechanisms: passwords, DSA/RSA keys

Important differences between SFTP and FTPS (cont. 1)

- **TCP/IP socket usage:**
  - FTP and FTPS - one “control” connection (port 21)
    - One “data” connection for each file transfer or directory listing.
    - Data connection is either setup server->client or client->server (“passive” mode) using dynamically assigned ports.
    - Can be troublesome for firewalls and NAT routers
  - SSH – one or more application “channels” are multiplexed in a single TCP/IP socket connection.
    - More “firewall/router friendly”

Important differences between SFTP and FTPS (cont. 2)

- **MVS dataset support:**
  - ✔ FTPS (IBM Comm Svr) - including load module libraries
  - ✗ SFTP (IBM Ported Tools)
  - ✔ SFTP (Co:Z)
  - ✓ SFTP (SSH Tectia) - “staged” and limited to 2GB unless partner is also Tectia

- **SMF (type 119) accounting:**
  - ✔ FTPS (IBM Comm Svr)
  - ✗ SFTP (IBM Ported Tools)
  - ✔ SFTP (Co:Z)
  - ✔ SFTP (SSH Tectia)

Important differences between SFTP and FTPS (cont. 3)

- **z/OS hardware crypto exploitation:**

  - **IBM Comm Svr FTPS**
    - ✔ Random number (entropy)
    - ✔ SAF/RACF key operations
    - ✔ Ciphers

  - **Ported Tools OpenSSH**
    - ✔ Random number (entropy) – via /dev/random with ICSF
    - ✔ SAF/RACF key operations - with Co:Z SFTP
    - ✗ Ciphers - *cards and letters to IBM please!*

  - **SSH Tectia for z/OS**
    - ✔ Random number (entropy)
    - ✔ SAF/RACF key operations
    - ✔ Ciphers

Important differences between SFTP and FTPS (cont. 4)

- **User Exits:**
  - Commonly used by customers or vendor products to control and automate file transfer operations.

✔ FTPS (IBM Comm Svr)
✗ SFTP (IBM Ported Tools)
✔ SFTP (Co:Z) – Support for IBM FTP compatible exits
✗ SFTP (SSH Tectia)

Managing FTP (and SFTP)

FTP/WatchDog-Z (SoftwareAssist.net)

An integrated product that manages z/OS FTP and Co:Z SFTP.

- Preemptive control over server usage via SAF/RACF rules
- Real-time monitoring of activity
- Automation and alert capabilities
- Comprehensive auditing of FTP and SFTP usage in minutes

Co:Z SFTP's compatibility with IBM FTP's user exits and SMF records allow it to be managed alongside FTP.

Which should I use – SFTP or FTPS?

- FTPS generally has better native z/OS features
  - SFTP MVS dataset support is available with Co:Z or SSH Tectia
  - SFTP User Exits are available with Co:Z
- FTPS likes PKI (X.509) – (do you?)
  - SSH Tectia also supports X.509 as a non-standard extension
  - Co:Z SFTP supports z/OS client authentication via SAF/RACF
- SFTP is more firewall/router friendly
- SFTP is more widely deployed on Unix/Linux
- SFTP generally has fewer incompatibilities between implementations

➔Your partners may dictate - the answer is often “both”

Using z/OS Ported Tools SFTP server

- Install and configure z/OS OpenSSH per the IBM manual
  - Create host keys
  - Use /dev/random and ICSF if possible!
  - Start SSHD (the OpenSSH server)

- How does SSHD find sftp-server subsystem?

```
# /etc/ssh/sshd_config
...
#Subsystem sftp  /usr/lib/ssh/sftp-server
# for using Co:Z SFTP -
Subsystem sftp /usr/local/coz/bin/sftp-server.sh
```

Using z/OS Ported Tools SFTP server

- From a non-z/OS OpenSSH sftp client:

```
kirk@ubuntu:~$ sftp kirk@zoshost
The authenticity of host 'zoshost (192.168.0.12)' can't be established.
RSA key fingerprint is 76:34:22:42:15:d6:f5:6e:82:61:d9:3c:00:13:12:ed.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'zoshost,192.168.0.12' (RSA) to the list
of known hosts.
kirk@zoshost's password: xxxxxx
sftp>
sftp> get zos_file local_file
```

- Under the covers, sftp uses the ssh command to connect to z/OS SSHD's sftp subsystem.
- Host key was accepted and added to the client file:

  `~/.ssh/authorized_keys`
- ssh option “`-o StrictHostKeyChecking=no`” will automatically accept a **new** host key

Using Co:Z SFTP server example

- IBM Ported Tools sshd_config sftp subsystem points to Co:Z sftp-server.
- From a non-z/OS sftp client:

```
kirk@ubuntu:~$ sftp kirk@zoshost
kirk@zoshost's password: xxxxxx
sftp> ls /+recfm=fb,lrecl=80
sftp> ls /+space=cyl.3.1
sftp> cd //KIRK
sftp> put local_file test.dsn
Uploading local_file to //KIRK/test.dsn
sftp> ls -al
Volume   Referred  Ext  Tracks    Used Recfm Lrecl BlkSz Dsorg  Dsname
VOL001  2009/08/04   2      45      18  FB      80 27920  PS    KIRK.TEST.DSN
VOL002  2009/02/10   1       1       1  U        0  6144  PS    KIRK.TEST.FOO
```

The z/OS Ported Tools sftp client in a batch job

```
//  EXEC PGM=BPXBATCH,PARM='SH /path/sftp-ex1.sh'
//STDOUT DD SYSOUT=*
//STDERR DD SYSOUT=*
//
```

(file: sftp-ex1.sh with “execute” bits set)

```
#!/bin/sh
sftp -b- kirk@myco.com <<EOB
get remote.file /path/local.file
EOB
```

- How is the userid and remote host authenticated?
- Additional steps to copy HFS/zFS files to/from datasets

The Co:Z SFTP client in a batch job

```
// EXEC  PGM=COZBATCH,  -- a better BPXBATCH
//       PARM='/rf=&RFILE ru=&RUSER rh=&RHOST'
//STDOUT DD SYSOUT=*
//STDERR DD SYSOUT=*
//DOWNLD DD DISP=(NEW,CATLG),DSN=..,DCB=...,SPACE=...
//STDIN  DD *  -- input to user's default login shell
ssh_opts=”-oStrictHostKeyChecking=no”
cozsftp $ssh_opts -b- $ru@$rh <<EOB
get $rf //DD:DOWNLD
EOB
//
```

- JCL/PROC variables substituted into environment variables
- Downloads a remote file into a dataset via DD reference
- Assumes user public key in remote `~/.ssh/authorized_keys`

Using a password from a batch sftp client

```
... (as previous slide) ...
//STDIN  DD *
export PASSWD_DSN='//HLQ.PASSWD(SITE1)'
export SSH_ASKPASS=read_passwd_dsn.sh
export DISPLAY=none
ssh_opts="-oBatchMode=no -oStrictHostKeyChecking=no"

cozsftp $ssh_opts -b- $ru@$rh <<EOB
get $rf //DD:DOWNLD
EOB
//
```

- Allows the use of a password from a RACF protected MVS dataset, and the acceptance of a ***new*** remote host key

Using a SAF/RACF Client Certificate

```
// EXEC  PGM=COZBATCH,  -- a better BPXBATCH
//       PARM='/rf=&RFILE ru=&RUSER rh=&RHOST'
//STDOUT DD SYSOUT=*
//STDERR DD SYSOUT=*
//DOWNLD DD DISP=(NEW,CATLG),DSN=..,DCB=...,SPACE=...
//STDIN  DD *  -- input to user's default login shell
ssh_opts=”-oStrictHostKeyChecking=no”
cozsftp $ssh_opts -k MY-RING -b- $ru@$rh <<EOB
get $rf //DD:DOWNLD
EOB
//
```

- MY-RING is the name of the user's SAF key ring
- The RSA private key from the client certificate will be used to sign the SSH client authentication request.

Setting up logon keys for z/OS sftp client (part 1)

```
zoshost:/u/kirk> mkdir .ssh; chmod 700 .ssh; cd .ssh
zoshost:/u/kirk/.ssh> ssh-keygen -t dsa
Generating public/private dsa key pair.
Enter file in which to save the key (/u/kirk/.ssh/id_dsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /u/kirk/.ssh/id_dsa.
Your public key has been saved in /u/kirk/.ssh/id_dsa.pub.
The key fingerprint is:
85:03:2d:99:10:19:2a:13:90:16:06:b6:7a:9b:e2:5c KIRK@ZOSHOST
```

- This needs to be done from a z/OS ssh session: ssh commands don't work in TSO OMVS.
- Consider using ACLs to secure ~/.ssh files from ***any access*** other than the owning userid

Setting up logon keys for z/OS sftp client (part 2)

```
zoshost:/u/kirk/.ssh> sftp kirk@myco.com
Connecting to myco.com...
The authenticity of host 'myco.com(192.168.0.15)' can't be
established.
RSA key fingerprint is
4d:d0:91:8b:5c:68:94:92:0b:6a:ec:b8:42:8e:fc:b6.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'myco.com,192.168.0.15' (RSA) to
the list of known hosts.
kirk@myco.com's password: xxxxxx
sftp>
```

- Now remote host's public key is in /u/kirk/.ssh/known_hosts

Setting up logon keys for z/OS sftp client (part 3)

```
(zoshost's sftp client still connected to remote host)
   sftp> pwd
   Remote working directory: /home/kirk/
   sftp> mkdir .ssh  (if necessary)
   sftp> chmod 700 .ssh
   sftp> cd .ssh
   sftp> ascii
   Sets the file transfer type to ASCII.
   sftp> put id_dsa.pub  authorized_keys
   sftp> chmod 600 authorized_keys
   sftp> quit
```

- Now z/OS client `known_hosts` has remote host's public key *and* remote host `~/.ssh/authorized_keys` has z/OS user's public key

```
zoshost:/u/kirk> sftp kirk@myco.com
sftp>
```

Common Pitfalls

- z/OS client or server userid must have an OMVS segment.
- If multiple z/OS userids share the same uid number, Ported Tools ssh and sshd won't necessary use “your” .ssh directory for keys
- SSH key files must be in EBCDIC on z/OS.
- Avoid ssh-rand-helper! Use /dev/random with ICSF if possible.
- Must use proper file permissions (or ssh may ignore your key files):

```
~/.ssh   -  700
  id_dsa, id_rsa (private keys) – 600
  authorized_keys – 600
  known_hosts  - 644
```

Trouble Shooting

- When debugging batch SFTP client job connection problems, test by using the interactive sftp client (or cozsftp) in an z/OS ssh shell using the same z/OS userid.
- Add “-vvv” option to OpenSSH sftp or ssh client to debug connection problems. Helps to compare log with similar working connection.
- Co:Z SFTP server has a per-session log file.   Tracing can be enabled in ~/.ssh/sftp-server.rc
- Consider setting up a test sshd server (see Co:Z SFTP Guide)
- See also: *IBM Ported Tools for z/OS User's Guide: “Trouble Shooting”*
- Post a question on our forum: http://dovetail.com/forum (see our support page for signup info)

Legal...

- Co:Z® is a registered trademark of Dovetailed Technologies
- SecureCRT® and SecureFX® are trademarks of Van Dyke Software Inc.
- SSH®, Secure Shell®, and TECTIA® are trademarks of SSH Communications
- z/OS® is a registered trademark of IBM Corporation