IBM TRAINING

E37

What Mother Never Told You about FTP on VSE

Don Stoever

IBM
SYSTEM z9 AND zSERIES EXPO
October 9 - 13, 2006

Orlando, FL

RETURN TO INDEX

What Mother Never Told You about FTP on VSE

Connectivity Systems

Product Development

Don Stoever

RFC 959
The File Transfer Protocol

- The objectives of FTP are:
  - to promote sharing of files and encourage use of remote computers
  - to shield a user from variations in file storage systems among hosts
  - to transfer data reliably and efficiently

RFC 959
The File Transfer Protocol

- Protocol is a set of rules
- Following the rules allows totally different systems to talk to each other

FTP Clients

- All ftp transfers have a single client, also referred to as the control connection
- FTP clients use the telent protocol to send commands and receive replies to a local and foreign FTP server

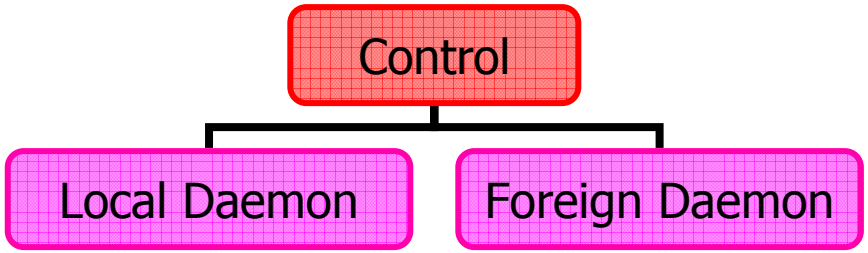
FTP Clients

- Examples of clients
  - // EXEC FTPBATCH
  - // EXEC FTP
  - WS_FTP Pro
  - MS-DOS FTP command
  - VM FTP command

FTP Clients

- Client opens connection to a:
  - Local FTP server(daemon)
  - Foreign FTP server(daemon)
  - Both usually require a userid and password
  - Clients often:
    - mask actual commands(DIR=LIST)
    - issue commands to each server at the same time
  - Both the foreign and local servers must support the standard set of commands as defined in RFC959

FTP Protocol



FTP Server commands

- ABOR
- ACCT
- ADAT
- ALLO
- APPE
- AUTH
- CDUP
- CONF
- CWD
- CWDX
- DELE
- ENC
- EVNT
- FEAT
- GEOJ
- HELP
- LIST
- MIC
- MKD
- MODE
- NLST
- NOOP
- PASS
- PASV

FTP Server commands

- PBSZ
- PORT
- PROT
- PWD
- PWDX
- QUIT
- REIN
- REST
- RETR
- RMD
- RNFR
- SITE
- SMNT
- STAT
- STOR
- STOU
- STRU
- SYST
- TYPE
- USER
- XCWD
- XMKD
- XPWD
- XRMD

FTP Clients

- FTP servers must return a 3 digit number as a reply(s) to a command
- Example // EXEC FTPBATCH
  - LUSER DON
    - The client ftpbatch sends
      - USER DON
    - telnet string to the local ftp server on VSE
  - The VSE ftp server replies with a:
    - 331 User name okay, need password

FTP DIR

- DIR
  - F: PORT 68,77,189,205,16,5
  - F: 200 PORT command okay
  - F: LIST
  - F: 150 File Listing Follows in ASCII mode.
  - F: 226 Transfer finished successfully.

FTP Put

- PUT
  - %SAM0,SAM,FB,80,800 FTPBSAM0.TXT
- L: PASV (VSE is passive end)
  - L: 227 Entering Passive Mode
  - (068,077,189,205,016,038).
- F: PORT 68,77,189,205,16,38
  - F: 200 PORT command okay
  - Tells foreign to issue active open to VSE
- I hate Firewalls and NATs!

FTP Put

- FTP910I Data connection open FTPBATCH,66.193.91.153(20)
- Data connection now is open…
- Command control connection waiting for 226 replies to RETR and STOR commands

FTP Put

- F: STOR FTPBSAM0.TXT
  - F: 150 "/AAAJUNK/FTPBSAM0.TXT" file ready to receive in ASCII mode
- IPF100I Sequential I/O Handler Startup

FTP Put

- L: RETR %SAM0,SAM,FB,80,800
  - L: 150-About to open data connection
  - File: Local.File.Definition
  - Type: ASCII Recfm: FB Lrecl: 80 Blksize: 800
  - CC=ON UNIX=OFF RECLF=OFF TRCC=OFF CRLF=ON
  - Translate with OS_02
  - 150 File status okay; about to open data connection

FTP Put

- FTP936I Sent 8001 bytes for file Local.File.Definition
  - L: 226-Bytes sent: 656,082
  - Records sent: 8,001
  - Transfer Seconds: 5.03 (128K/Sec)
  - File I/O Seconds: .39 (640K/Sec)
  - 226 Closing data connection.
  - F: 226 Transfer finished successfully.

FTP Put

- ACTIVE put in before PUT:
  - F: PASV
  - F: 227 Entering Passive Mode (66,193,91,153,140,64)
  - L: PORT 66,193,91,153,140,64
  - L: 200 Command okay
  - F: STOR FTPBSAM0.TXT
  - FTP321W Connection stalled at 12:01:40
  - F: 425 Timeout on network transaction - closing connection.

FTP Data Connection

- Type
  - Image
    - Binary data
  - ASCII
    - Text files
    - Lines end with CR/LF x0A0D
  - EBCDIC
    - Text files
    - Lines end with NL x15

FTP Data Connection

- Mode
  - Stream
    - Data is stream of bytes
  - Block
    - Data is series of blocks with headers
  - Compressed
    - Supported never seen used

FTP Data Connection

- Structure
  - File
    - Default
    - Data is continuous bytes
  - Record
    - Data is framed

Mode Stream File

- Stream File
  - Image
    - No control codes
    - End of file is connection close
  - ASCII
    - End of record is CRLF
    - End of file is connection close
  - EBCDIC
    - End of record is NL
    - End of file is connection close

Mode Stream Record

- Stream Record
  - Image/Binary
    - Escape char is x'FF'
    - Next byte is meaning:
      - 1111 1111 - Single data byte
      - 0000 0001 - EOR
      - 0000 0010 - EOF
      - 0000 0011 - EOR & EOF

Mode Stream Record

- Stream Record
  - ASCII
    - Escape char is x'FF'
    - Next byte is meaning:
      - 1111 1111 - Single data byte
      - 0000 0001 - EOR
      - 0000 0010 - EOF
      - 0000 0011 - EOR & EOF

Mode Stream Record

- Stream Record
  - EBCDIC
    - Escape char is x'FF'
    - Next byte is meaning:
      - 1111 1111 - Single data byte
      - 0000 0001 - EOR
      - 0000 0010 - EOF
      - 0000 0011 - EOR & EOF

Mode Block

- Block File: Not supported
- Block Record Image
  - 3-byte header:
  - 1 byte: Type
    - 1... .... - EOR
    - .1.. .... - EOF
    - ..1. .... - Unreliable (ignored)
    - ...1 .... - Restart marker
  - 2 bytes: Length
  - Data follows

Mode Block Record

- Block Record EBCDIC
  - 3-byte header:
  - 1 byte: Type
    - 1... .... - EOR
    - .1.. .... - EOF
    - ..1. .... - Unreliable (ignored)
    - ...1 .... - Restart marker
  - 2 bytes: Length
  - Data follows

Mode Block Record

- Block Record ASCII
  - 3-byte header:
  - 1 byte: Type
    - 1... .... - EOR
    - .1.. .... - EOF
    - ..1. .... - Unreliable (ignored)
    - ...1 .... - Restart marker
  - 2 bytes: Length
  - Data follows

FTPBATCH as a External Server

- Advantages:
  - Moves all file opens/closes to external partition
  - All I/O done outside of TCP/IP
  - Free's up tcp/ip to focus on network
  - Improved recoverability
  - Support for dataspaces
  - Prioritize workloads

FTPBATCH as a External Server

- // JOB FTPB0021
- // OPTION LOG,PARTDUMP
- // OPTION SYSPARM='00'
- * * Assign's, Dlbl's, Extent's for all defined files
  - Should be taken from TCP/IP jcl

- // EXEC FTPBATCH,SIZE=FTPBATCH,
  - PARM='FTPDPORT=21'

FTPBATCH as a External Server

- // EXEC FTPBATCH,PARM='PARMS'
  - FTPDPORT=0021
  - UNIX=BIN
  - MAXACT=10
  - WELCOME=WMVSEDRS
  - DYNFILE=NO
  - ABORT=YES

FTPBATCH as a External Server

- FTPBATCH sysipt commands:
  - SET EXTTYPES NO
  - SET DIAGNOSE EVENTS
  - SET DIAGNOSE ON
  - SET IDLETIME 36000
    - Divide by 300=seconds to terminate idle sessions
  - SET TERSE ON
  - SET PULSE OFF
    - Disables data connection pulsing

FTPBATCH as a External Server

- FTPBATCH sysipt commands:
  - MSGSUPP FTP910
  - SET MSGXLOG ON
  - SET DATAWECB ON
  - SET SENDFAST ON
  - SET FIOWAIT ON (HFS only)
  - SET NULLRECD NOTHING

FTP Automatic Security

- No Need to code a security exit!!!
  - SECURITY ON
    - AUTO=ON
    - BATCH=ON
    - MODE=FAIL LOGGING=FAIL

FTP Automatic Security

- New ASECURITY command:
  - ASECURITY FTPD=YES FTPC=YES
  - ASECURITY BLOCKIP=YES
  - ASECURITY BLOCKCNT=3
- TRUST ADD IP=66.193.91.130
- ACCESS CLEAR
- ACCESS CLEAR IP=

FTP Automatic Security

- DEFINE USER operands to create a
  - FTP READ ONLY user...
    - DEFINE USER
    - ID=CSIVSEDR,PASSWORD=READ2357
    - DATA=YYNNNNNNYNNNNNYYYYNNNNNNNNNYNNNYNNNNNNNNNNN
    - ROOT='/HFS001/CSIVSEDR',FTP=YES

FTP Automatic Security

- DATA=YYNNNNNNYNNNNNYYYYNNNNNNNNNY NNYNNNNNNNNNN
  - SXTYPASS EQU 1 - Password Check
  - SXTYREAD EQU 2 - Read Check
  - SXTYWRIT EQU 3 - Write Check
  - SXTYUPDT EQU 4 - Update Check
  - SXTYCMD EQU 9 - SITE Command check
  - SXTYDEL EQU 10 - Delete check
  - SXTYREN EQU 11 - Rename check
  - SXTYCRT EQU 12 - Create check
  - SXTYEXEC EQU 13 - EXEC command check

FTP Automatic Security

- DATA=YYNNNNNNYNNNNNYYYYNNNNNNNNY NNYNNNNNNNNNN
  - SXTYAPPE EQU 14 - APPEND check
  - SXTYOPDI EQU 15 - OPDIR check
  - SXTYRDD EQU 16 - RDDIR check
  - SXTYCWD EQU 17 - CWD Check
  - SXTYLOGI EQU 20 - Daemon LOGIN request
  - SXTYMKD EQU 24 - Make directory
  - SXTYRMD EQU 25 - Remove directory
  - SXTYCWDL EQU 26 - Last CWD
  - SXTYFCMD EQU 29 - FTPD command

FTP Automatic Security

- Suppose you want to stop any new ftp sessions from being established on VSE
- Simply issue a:
- ASECURITY FTPD=NO
  - No 220-welcome to VSE msg will be sent out to anyone connecting into VSE on the ftp port(usually 21)

HFS Encrypted Files

- File can be stored on VSE with FTP encrypted!!!
  - Simply use the DEFINE FILE command
  - DEFINE FILE
    - DLBL=HFSTST,
    - PUBLIC=HFSTST
    - TYPE=HFS,RECFM=S,LRECL=4096
    - CIPHER=SDESCBCSHA1
    - CIPHERKEY=CIALHFSK

HFS Encrypted Files

- File can be stored on VSE with FTP weak or strong cryptography and hashing for integrity
  - CIPHER=NULL-NULL
  - CIPHER=SDESCBC-SHA1
    - Single DES
  - CIPHER=TDESCBC-SHA1
    - Triple DES
  - CIPHER=AES128C-SHA1
    - Rjindel

HFS Encrypted Files

- Allows complete control of keys and ciphers used
  - CIPHERKEY=CIALHFSK
  - CIPHERKEY=user_defined
  - CIPHER=KEYMASTER

FTP Security and Integrity

- Transmits commands, responses, and data in the clear with no:
  - Authentication
  - Privacy
  - Integrity
- Hey, wait a minute aren't the FTP USER and PASS commands good enough?
- What about a truncation attack?

FTP Security and Integrity

- So how can I ?
  - Authenticate sender/receiver
  - Guarantee Privacy of confidential data
  - Guarantee Integrity of the data

Secure FTP

- Internet Engineering Task Force(IETF) draft document:
- "Securing FTP with TLS"
- Widely accepted de-facto standard for securely tranmitting files with the FTP protocol.

Secure FTP

- Secure FTP provides:
  - User authentication
  - Privacy
  - Integrity
- By using industry standard cryptographic functions :
  - RSA digitally signed certificates
  - DES encryption
  - SHA-1 secure hash functions.

Secure FTP

- Protection for commands and data transmitted for the FTP protocol
- By implementing the SSL protocol for FTP clients and servers running on the VSE platform
- Secure FTP implements both the SSL 3.0 and TLS 1.0 standards for security

Secure FTP

- Allow interoperation across platforms
  - RFC-959 defines the FTP protocol
  - RFC-2228 FTP Security Extensions
  - RFC-2389 Feature Negotiation Mechanism for FTP
  - RFC-2246 defines the TLS protocol
  - RFC-2577 FTP Security Considerations

Secure FTP

- New FTP commands:
  - FEAT
  - AUTH
  - PBSZ
  - PROT

Secure FTP

- FEAT command
  - RFC2389 allows clients to find out what features the FTP daemon supports
  - 211-Extensions supported
    - AUTH SSL
    - PBSZ
    - PROT
  - 211 END

Secure FTP

- AUTH SSL command
  - Issued by client
  - Causes a SSL session to be negotiated
  - Must be first command after OPEN
    - All other commands rejected until SSL enabled FTP daemon gets this!
  - Protects the control/command connection to the foreign FTP daemon
  - AUTH TLS also allowed as synonym
  - SSL is self-negotiating…

Secure FTP

- PBSZ command
  - RFC2228 Protection Buffer Size
  - Required Prior to PROT command
  - Not coded by end-user
    - Like when you do a PUT, internally FTP issues PORT, RETR, STOR
  - Not really used for anything but is still required…because…

Secure FTP

- RFC2246 TLS/SSL protocol max buffer size is 32k, because…
  - has 2 byte length in its record header
  - Cryptos use block ciphers DES-CBC, etc.
  - DEFINE FTPD transfer buffer size
    - 1.4A-E = 32k shared buffers
    - 1.5A = 128k shared buffers
    - 1.5B = 64k dedicated buffers
      - But user defineable, BUFCNT=, BUFSIZE=

Secure FTP

- PROT command
  - Defines security for the data connection
  - You can just secure command connection
  - Data Connection can be:
    - PROT C – Clear No Privacy or Integrity
    - PROT P – Private Privacy and Integrity
    - PROT S – Safe No Privacy, but Integrity
    - PROT E – Confidential Privacy, but no Integrity

Secure FTP

- All controlled be Server Policy that may:
  - Deny any commands before SSL negotiation
  - Define level of SSL/TLS to be used
  - Define cipher suites to be used
  - Allow SSL/TLS client authentication instead of USER/PASS, or require both!
  - Insist on data connection security

Questions?

RETURN TO INDEX