FTP Analysis via SMF Records, FTP Server Exits and Logging, and CTRACE

David J Cheng

Applied Expert Systems, Inc.

davec@aesclever.com

March 15, 2010, 4:30PM
Session 3316

SHARE in Seattle

Agenda

- FTP vs. SFTP vs. FTPS
- FTP Background
  - Data type, structure and mode
  - *Active* FTP vs. *Passive* FTP
  - FTP Commands and Replies
- FTP Diagnostic/Performance Data
- FTP Server Exits – logging for audit trail
- FTP SMF Records – Server, Client
- TCP Connection SMF Records – INIT, TERM
- SMF Exits vs. NMI SYSTCPSM
- FTP Server Logging
- Tracing
  - FTP Server Trace
  - IP packet trace, OSAENTA trace
  - Packet Trace Analysis
- FTP Diagnosis, Tuning and Analysis

FTP vs. SFTP vs. FTPS

FTP

- “regular” or “normal” File Transport Protocol
- Use one port (21) for commands and another port (e.g., 20) for data transfer
- RFC 959 (October 1985)
- RFC 1579 – Firewall-Friendly FTP (PASV)

SFTP

- SSH FTP or Secure FTP
- Use a single port (22) to multiplex commands and data transfer
- It’s the default FTP server for Linux

FTPS

- FTP Secure, or FTP-SSL, or AT-TLS FTP
- Adds support for TLS (SSL) for encryption
- RFC 2246 (TLS 1.0)
- RFC 2228 – FTP Security Extensions (AUTH)

FTP Data Type – how data is interpreted by the receiver

- FTP always transfer data in 8-bit bytes; this is called the *transfer size*
  - ASCII (**TYPE A**) - default data type
    - Strip off local line terminators
    - Each line of data is terminated by CRLF (X'0D0A')
    - =====➔
    - Strip off CRLF
    - Add local line terminators
    - Translation is always required; even between 2 ASCII hosts:
      ASCII -> NVT-ASCII -> ASCII
      (NVT-ASCII : Network Virtual Terminal ASCII as defined in the TELNET protocol.)
    - If MVS is the receiving side, data will be translated to EBCDIC and CRLF replaced with MVS record boundaries – according to SITE/LOCSITE parms: RECFM and LRECL
  - EBCDIC (**TYPE E**)
    - 8-bit EBCDIC bytes are transferred as they are – no translation
  - IMAGE (**TYPE I**)
    - Contiguous bits packed into the 8-bit FTP transfer byte size
    - Normally used for binary data
    - More efficient method to transfer data between 2 similar ASCII hosts

FTP Data Structure – how data is stored by the receiver

- File (STRU F)
  - Has no internal structure
  - Contiguous sequence of bytes
  - Most widely implemented
- Record (STRU R)
  - File is made up of sequential records; ASCII type with CRLF sequences
  - z/OS only supports Record structure with *stream* mode transfer
- Page (STRU P) – not supported in z/OS

FTP Data Mode – how data is transmitted

- Stream (MODE S)
  - Transmitted as stream of bytes; with very little or no extra processing
- Block (MODE B)
  - Transmitted as a series of data blocks, each block is preceded by a header - count and descriptor
  - z/OS only supports Block mode with data type EBCDIC
  - A file transferred between 2 z/OS systems in Block mode will preserve its record structure (e.g., variable length records)
- Compress (MODE C)
  - Transmitted in a compressed format
  - Simple compression algorithm – send duplicated bytes in a two-byte sequence, followed by a one-byte filler
  - In z/OS Compress requires data type EBCDIC

Control / Data Connections

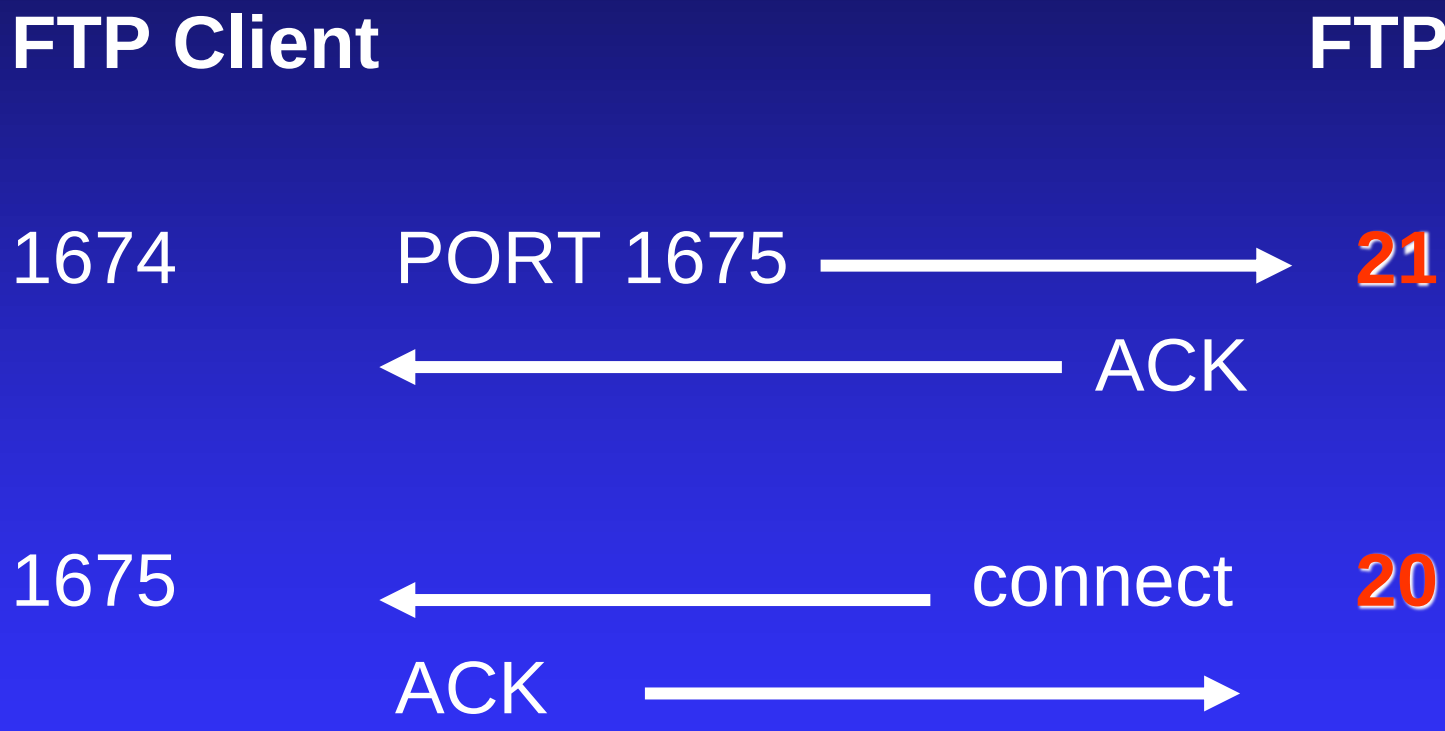
- ***Control* connection**
  - A communication path between the Client and Server for the exchange of commands & replies
  - FTP Server Port 21
  - Connection stays up during the whole session, in which many files may be transferred
- ***Data* connection**
  - A full duplex connection over which data is transferred, in a specified mode and type
  - FTP Server Port 20 (for active FTP)
  - Usually one for each file transfer

Active FTP

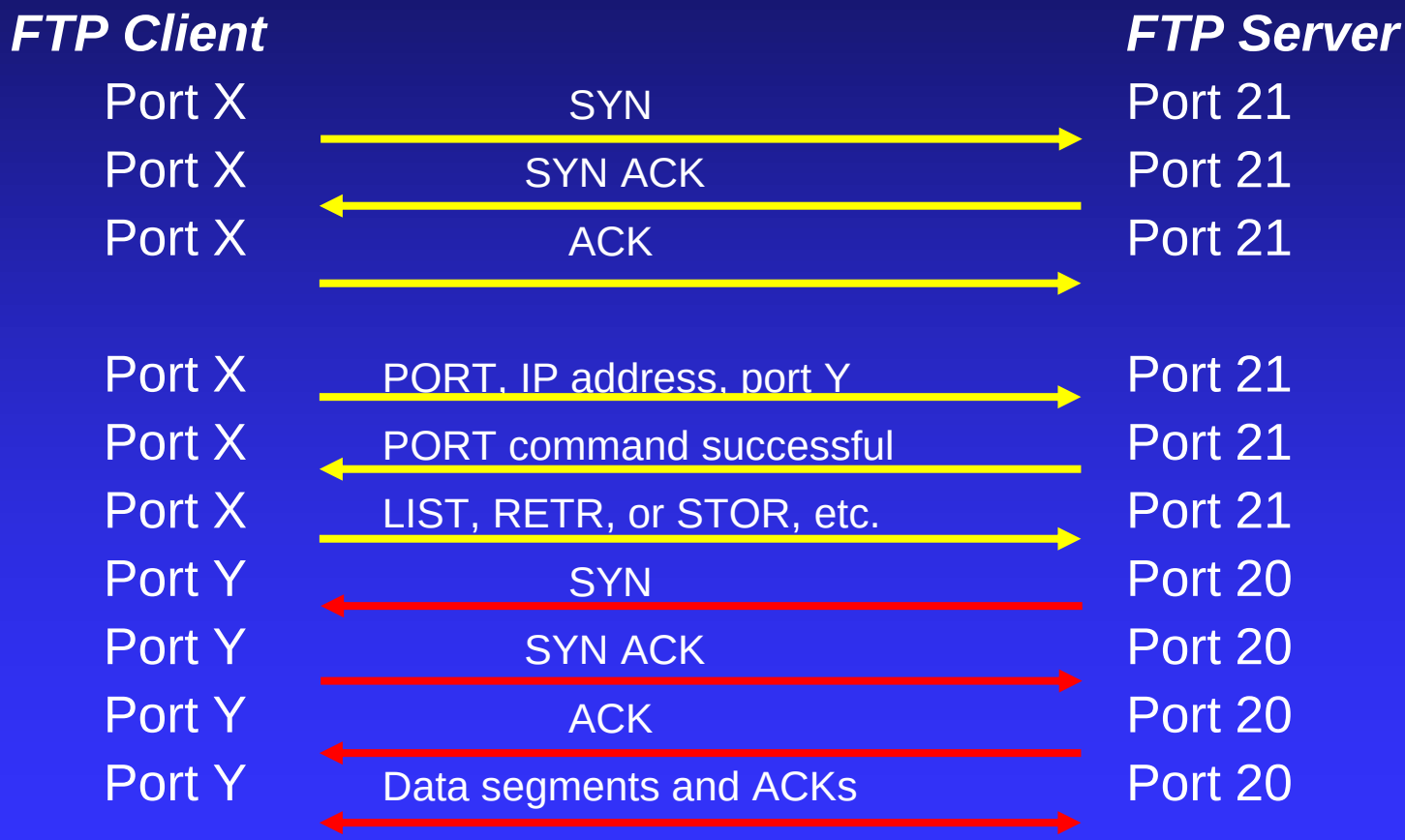
- Server initiates *data connection* to the client
- Client connects from a random unprivileged port (N > 1024) to the FTP server's port 21
- Client starts listening to port N+1 and sends the FTP command PORT N+1 to the FTP server
- PORT h1,h2,h3,h4,p1,p2

  h1,h2,h3,h4 is the client's IP address, p1,p2 is the client port number in an 8 bit high, low bit order
- The Server will then connect back to the client's specified data port from its local data port (port 20)

Active FTP



FTP Active Mode in Detail

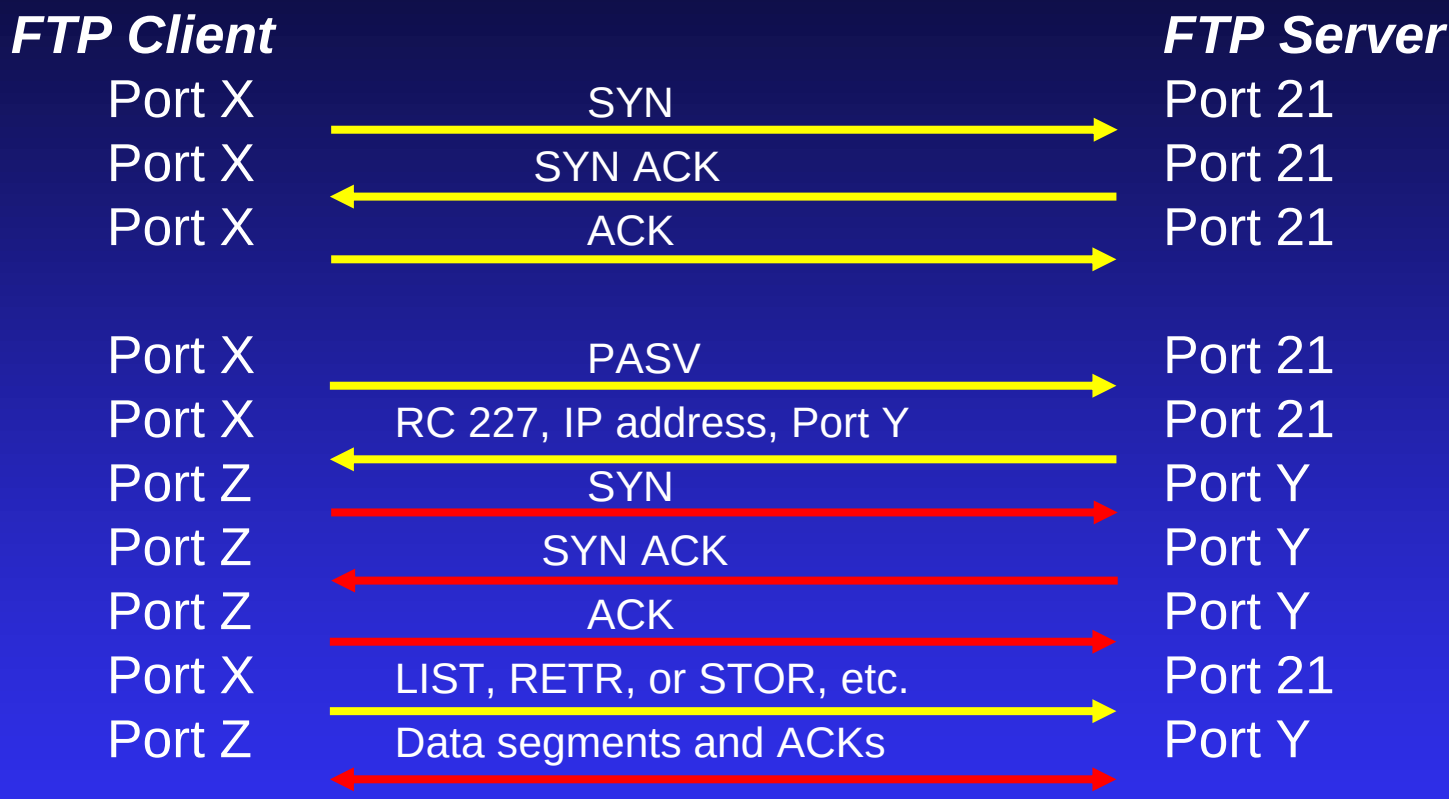


Passive FTP

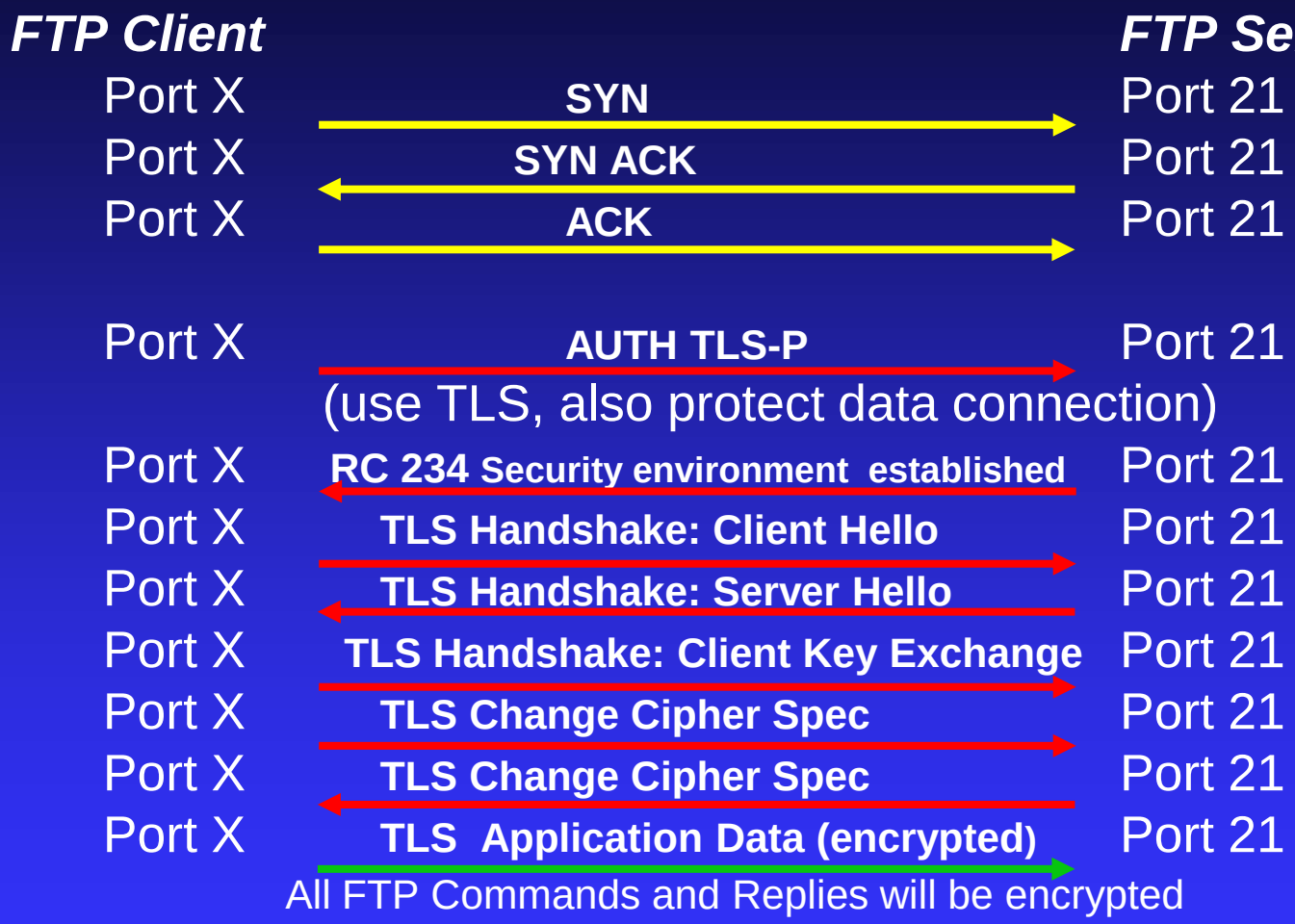
- Client initiates data connection to the server
- Firewall friendly
- When opening an FTP connection, the client opens 2 random unprivileged ports locally (N > 1024 and N+1)
- The first port contacts the server on port 21
- Client issues the PASV command (the PASV command takes no parameters)
- The server then opens a random port and sends Reply Code 227 back to the client (similar to the PORT command)
- The client then initiates the connection from port N+1 to port P on the server to transfer data

Passive FTP

FTP Passive Mode in Detail



FTPS Flow - Control Connection



FTP Commands

- Commands and Replies are sent across the control connection and are in plain text.
- Commands are 3 or 4 bytes characters, each with optional parameters.
- *The FTP commands specify the parameters for:*
  - the data connection (port)
  - transfer mode
  - data representation type and structure
  - the nature of file system operation (store, retrieve, append, delete, etc.)

Sample FTP Commands

Access Control:

- ABOR — Abort a file transfer
- USER — User identification
- PORT — Data port specification
- QUIT — Terminates a user and the control connection

Transfer:

- TYPE — Data representation (ASCII, EBCDIC, Image)
- STRU — Transfer structure (File, Record, Page)
- MODE — Transfer mode (Stream, Block, Compress)
- RETR — Server -> Client file transfer
- STOR — Client -> Server file transfer

Service:

- DELE — Deletes a Server file
- LIST — Directory listing
- RNFR — Renames from <old file name>
- RNTO — Renames to <new file name>

(RNFR must be immediately followed by a RNTO command)

FTP Replies

- Synchronization of requests and actions in the file transfer process
- Guarantee that the user process always knows the state of Server
- Every command must generate at least one reply
- An FTP reply consists of a 3-digit number (i.e., 3 alphanumeric characters) followed by some text
- The number is intended for use by the software to determine what to do next; the text is intended for the human user
- There may be more than one reply, in which case these multiple replies must be easily distinguished

FTP Reply Code

- 1yz Positive preliminary reply
- 2yz Positive completion reply (a new command may be sent)
- 3yz Positive intermediate reply (another command must be sent)
- 4yz Transient negative reply (command can be re- issued later)
- 5yz Permanent negative reply (command should not be retried)

yz range: 00 .. 59 (e.g., 100-159, 200-259, etc.)

- x0z Syntax error
- x1z Information requested by the client
- x2z Information relating to the control or data connection
- x3z Authentication and accounting
- x4z No in current use
- x5z File system status

Sample FTP Reply Codes

- 150 File status okay; about to open data connection.
- 226 Transfer complete (FTP.DATA: REPLY226 TRUE)
- 227 Entering passive mode {h1,h2,h3,h4,p1,p2}
- 250 Requested file action okay, completed.
- 257 "PATHNAME" created.
- 350 Requested file action pending further information.
- 450 Requested file action not taken. File unavailable (e.g., file busy).
- 550 Requested action not taken. File unavailable (e.g., file not found, no access).
- 451 Requested action aborted. Local error in processing.
- 551 Requested action aborted. Page type unknown.
- 452 Requested action not taken. Insufficient storage space in system.
- 552 Requested file action aborted. Exceeded storage allocation (for current directory or data set).
- 553 Requested action not taken.

```
C:\Windows>ftp 137.72.43.247
Connected to 137.72.43.247.
220-FTPD1 IBM FTP CS V1R7 at os17.aesclever.com, 21:05:48
    on 2006-07-20.
220 Connection will close if idle for more than 5 minutes.
User (137.72.43.207:(none)): p390
331 Send password please.
Password:
```

RC 220	**Service ready for new user**
	- : continuation
RC 331	**User name OK, need password**

FTP Vulnerabilities

- User ID and Password are sent in clear text
- FTP Bounce Attack
  - Arbitrary IP and port can be specified in the PORT command – this allows an attacker to open a (data) connection to a port on a machine that's not the original client
    - Port scanning
    - Bypass firewalls
- To Control PORT or EPRT command:
  - PORTCOMMAND ACCEPT | REJECT
  - PORTCOMMANDIPADDR UNRESTRICTED | NOREDIRECT
  - PORTCOMMANDPORT UNRESTRICTED | NOLOWPORTS (ports < 1024)

FTP Diagnostics/Performance Data

- Connection attempts
- Logon failures
- Client identification
- Active vs. Passive FTP
- FTP commands
- FTP replies
- Throughput

Data can also be used as ***audit trail*** and for monitoring security breaches.

FTP Diagnostics/Performance Data

- Published Record Types or API
- Non-intrusive, lower overhead
- Event-driven
  - True real-time data
  - FTP Server exits and SMF exits
- Waiting/”Polling”
  - Comm Server Network Management API (SMF 119 only)

FTP Diagnostics/Performance Data

- FTP Server Exits
  - FTCHKIP – open connection (invoked by FTP daemon)
  - FTCHKPWD – password verification
  - FTCHKCMD – FTP command
  - FTPOSTPR – FTP command completion
  - FTCHKJES – Job submission
  - FTPSMFEX – FTP server SMF record (“obsolete”)
- SMF records (Type 118 or 119)
  - FTP Server Logon Failure
  - FTP Server Transfer Completion
  - FTP Client Transfer Completion
  - TCP Connection Initiation
  - TCP Connection Termination

FTP Diagnostics/Performance Data

- Logging/Tracing
  - FTP Server
    - FTPLOGGING
    - ANONYMOUSFTPLOGGING
    - TRACE, DEBUG
  - Packet trace – detailed analysis at protocol level
    - NOT for monitoring purpose
    - Performance penalty (e.g., APAR PQ84192)

FTP Server User Exits

- R1 -> parameter list, which is a series of pointers to values
- The first word of the parameter list always points to the return code (RC). RC=0 upon entry to an exit. If RC is not 0, user will receive a negative reply
- The second word of the parameter list always points to a word containing the number of parameters that follow
- APF-authorized
- STEPLIB, Linklist, or LPA
- RACF consideration
- Sample code in TCPIP.SEZAINST

FTP Server Exit - FTCHKIP

FTCHKIP is called at the initial stage of login or whenever the user issues an OPEN command.

It is loaded at FTP daemon initialization - need to recycle the FTP server to use an updated version of FTCHKIP.

- Client's IP address (IPV4) and port
- Server's IP address (IPV4) and port
- Socket address structure (IPV4 or IPV6) for the client's control connection
- Socket address structure (IPV4 or IPV6) for the server's control connection
- Session ID

- Security Control: control FTP access by IP address

FTP Server Exit - FTCHKPWD

FTCHKPWD is called after the user enters the password

- Client's user ID
- Client's password
- User data
- **Number of bad passwords input in this logon attempt**
- Socket address structure for the client's control connection
- Socket address structure for the server's control connection
- Session ID

Security Control: Control FTP access by User ID, or invalid logon attempts

FTP Server Exit - FTCHKCMD

FTCHKCMD is called whenever the client enters a command

- Client's user ID
- Command
- Command parameters
- Current directory type: MVS, HFS
- File type: SEQ, JES, SQL
- Current working directory
- Address of a buffer for command modification

Security Control: Control FTP access by FTP command or file name, etc.

FTP Server Exit - FTPOSTPR

FTPOSTPR is called upon completion of the FTP commands RETR, STOR, STOU, APPE, DELE, and RNTO

- Client's user ID
- Client's IP address
- Client's port
- Current directory type: MVS, HFS
- Current working directory
- Current file type: SEQ, JES, SQL
- FTP reply code
- FTP reply string
- FTP command code
- Current CONDDISP setting: C for catalog, D for delete

FTP Server Exit - FTPOSTPR

- Close reason code:
  - 0 – transfer completed normally
  - 4 – transfer completed w/error
    see FTP reply code and text string
  - 8 – transfer completed w/socket erros
  - 12 – transfer aborted
  - 16 – transfer aborted w/SQL file errors
- Dataset name or HFS file name
- Bytes transferred
- Socket address structure for the client's control session
- Socket address structure for the server's control session
- Session ID
- Address of scratch pad area (256 bytes)

FTP Server Exit - FTPSMFEX

- FTPSMFEX is called before a type **118** SMF (FTP server) record is written to SMF
- Type 119 SMF records must use the system-wide SMF exits: IEFU83, IEFU84 , (IEFU85)
- R1 -> the following parameter list:
  - Pointer to the return code
  - Pointer to the type 118 SMF record
- On entry, the return code is set to 0. A return code of 0 specifies that the SMF record will be written

FTP Server Exit Installation

- APF-authorize the load library
- Add the load library to STEPLIB in the FTPD proc
- If RACF Program Control is active: SETROPTS WHEN(PROGRAM), you must define FTP exits to RACF class PROGRAM
- Restart the FTP Daemon (for FTCHKIP)

FTP Server Exit Installation

Sample RACF Definition for FTCHKIP:

RDEFINE PROGRAM FTCHKIP

ADDMEM('*loadlib*'/*volser*/NOPADCHK) UACC(READ)

…

SETR WHEN(PROGRAM) REFRESH

→ Without proper RACF definition, FTP client will get the following error when logging in:

550 PASS COMMAND FAILED - _PASSWD() ERROR: EDC5157I AN INTERNAL ERROR OCURRED

Verify FTP Server Exits

- Start the FTP Server with the “TRACE” parameter; e.g., **S FTPD,PARM=TRACE**
- Check for the following messages in SYSLOG:

```
BPXF024I (FTPD) Jan  5 18:01:34 ftpd 33619980 : DM1009 main:
   FTCHKIP successfully loaded
BPXF024I (AESDJC1) Jan  6 02:01:57 ftps 16843115 : RX0625 main:
   chkpwdexit successfully loaded
BPXF024I (AESDJC1) Jan  6 02:01:58 ftps 16843115 : RX0641 main:
   chkcmdexit successfully loaded
BPXF024I (AESDJC1) Jan  6 02:01:58 ftps 16843115 : RX0696 main:
   FTPOSTPR successfully loaded
```

FTP Server Transfer Completion SMF Record

- FTP command
- FTP type: SEQ, JES, SQL
- Client IP address and port
- Server IP address and port
- Local user ID
- Data format:
  - A: ASCII
  - E: EBCDIC
  - I: image (binary)
  - D: double byte
  - U: UCS-2
- Data Structure: File or Record
- Transmission mode – S: stream, B: block. C : compressed
- Start/End time of transmission
- Bytes transferred
- FTP reply code
- Dataset/member/file names
- File transmission duration

FTP Client Transfer Completion SMF Record

- FTP command
- Client IP address and port
- Server IP address and port
- Data format
- Data Structure
- Transmission mode
- Start/End time of transmission
- Byte count
- Dataset/file name
- File transmission duration

Enable TCP/IP SMF Recording

- SMFPRMxx – make sure that 118/119 is not being excluded from recording
- SMF Type 119 is available in z/OS V1R2 and later releases
- SMF Type 118 and Type 119 can co-exist
- To get FTP *Server* SMF record, configure *FTP DATA* as follows:
  - 118: ~~**SMF STD**~~
  - 119: **SMF TYPE119**

Enable TCP/IP Session SMF Recording

- To get FTP Client SMF record, configure *TCP/IP PROFILE* as follows:

  SMFCONFIG TYPE119 FTPCLIENT …

- To get TCP INIT and TERM SMF records:

  SMFCONFIG TYPE119 TCPINIT TCPTERM …

- Do not collect duplicate records; either 118 or 119 (recommended)

- TERM records are “better” than INIT records: more information

Verify SMF Recording

- System Level – issue the **D SMF,O** operator command, verify:
  - SMFPRMxx member
  - SMF parameters
- TCP/IP Level – issue the **NETSTAT,CONFIG** command
  - Check the SMF Parameters listing; e.g.,

```
SMF Parameters:
 Type 118:
   TcpInit:       00   TcpTerm:     00   FTPClient:     00
   TN3270Client:  00   TcpIpStats:  00
 Type 119:
   TcpInit:       Yes  TcpTerm:     Yes  FTPClient:     Yes
   TcpIpStats:    Yes  IfStats:     Yes  PortStats:     Yes
   Stack:         Yes  UdpTerm:     Yes  TN3270Client:  Yes
```

Verify SMF Recording

- **FTP Server** – start the FTP server with the "TRACE" parameter;
  e.g., **S FTPD,PARM=TRACE**
  - Look for the write_smf_record messages; e.g.,

```
250 Transfer completed successfully.
BPXF024I (AESDJC1) Jan  6 02:02:08 ftps 16843115 : RU1463
    write_smf_record: entered with type 4
BPXF024I (AESDJC1) Jan  6 02:02:08 ftps 16843115 : RU0754
    write_smf_record_119: entered with type 4.
```

- **FTP Client** – start the FTP client with the "trace" parameter, or issue the "debug" command from an FTP client session;
  e.g., ftp 137.72.43.247 (trace
  - Look for the following messages: CU1963, CU1463, CU2241; e.g.,

```
250 Transfer completed successfully.
EZA1617I 2320 bytes transferred in 0.160 seconds.  Transfer
    rate 14.50 Kbytes/sec.
CU1963 write_smf_record: entered with type 16.
CU1463 write_smf_record_119: entered with type 16.
CU2241 write_smf_record: length of smfrecord: 224
```

Obtaining SMF data in real-time – SMF Exits

- SMF Exits - receives control when caller invokes the macro to write SMF records; Supervisor state, KEY 0
  - IEFU83 – SMFWTM, BRANCH=NO
    - FTP Server SMF
    - TCP/IP Session SMF (INIT and TERM)
  - IEFU84 – SMFWTM, BRANCH=YES
    - FTP Client SMF
  - IEFU85 – SMFEWTM, BRANCH=YES MODE=XMEM, and ASCB != home primary ASID.

    IEFU85 cannot issue any SVC
  - MVS Dynamic Exits Facility allows multiple exits to co-exist

Installing SMF Exits

- Specify SMF Exits in SMFPRMxx
  - By default, ALL exits are invoked
  - If EXITS is specified on a SUBSYS for STC, OMVS, TSO or JES2, specify IEFU83, IEFU84 and IEFU85 on the EXITS parameter
  - E.g, SUBSYS(STC,EXITS(IEFU83,IEFU84,IEFU85,IEFACTRT))
- Specify SMF Exits to the Dynamic Exits Facility
  - Define PROGxx in SYS1.PARMLIB
    - EXIT ADD EXITNAME(SYS.IEFU83) MODNAME(*module*) DSNAME(*dsn*)
    - EXIT ADD EXITNAME(SYSSTC.IEFU83) MODNAME(module) DSNAME(dsn)
  - Issue the SET PROG=xx command
- Verify SMF Exits
  - **D PROG,EXIT**
  - **D PROG,EXIT,EX=SYS.IEFU83,DIAG**

```
CSV464I 21.37.38 PROG,EXIT DISPLAY 109
EXIT SYS.IEFU83
MODULE    STATE EPADDR    LOADPT    LENGTH    JOBNAME
IEFU83      A   838CA518  00000000  00000000  *
AESSMF00    A   8FFB66F8  0FFB66F8  00000908  *
```

Obtaining SMF data in real-time - NMI

- z/OS CS Network Management Interface (NMI)
  - SYSTCPSM interface
  - Type 119 SMF records:
    - FTP Server/Client, TN3270 Server and TSO Telnet Client
    - TCP Connection records
  - Requires waiting/"polling" – not as real-time as SMF exits

SYSTCPSM NMI

- SAF authorization for the SERVAUTH class **EZB.NETMGMT.sysname.tcpprocname.SYSTCPSM** profile
- TCP/IP Profile
  - NETMONITOR **ON** or
  - NETMONITOR … **SMFSERVICE**
- Connect to the AF_UNIX socket descriptor: /var/sock/SYSTCPSM.<tcpprocname>
- Receive the first INIT record
- Wait to receive token records from TCP/IP
  - A record will be sent when the buffer is full (max size 64K)
  - Record for partial buffer will be sent if there has been no activity “for a brief period”

SYSTCPSM Service

- Use the EZBTMIC1 service (in SYS1.CSSLIB) to copy the data into application buffer
- Caller must be APF authorized
- Input is the token record received from SYSTCPSM NMI
- Data is copied to application buffer as *CTE (Component Trace Element)* records:
  - <header, data, epilogue>,…

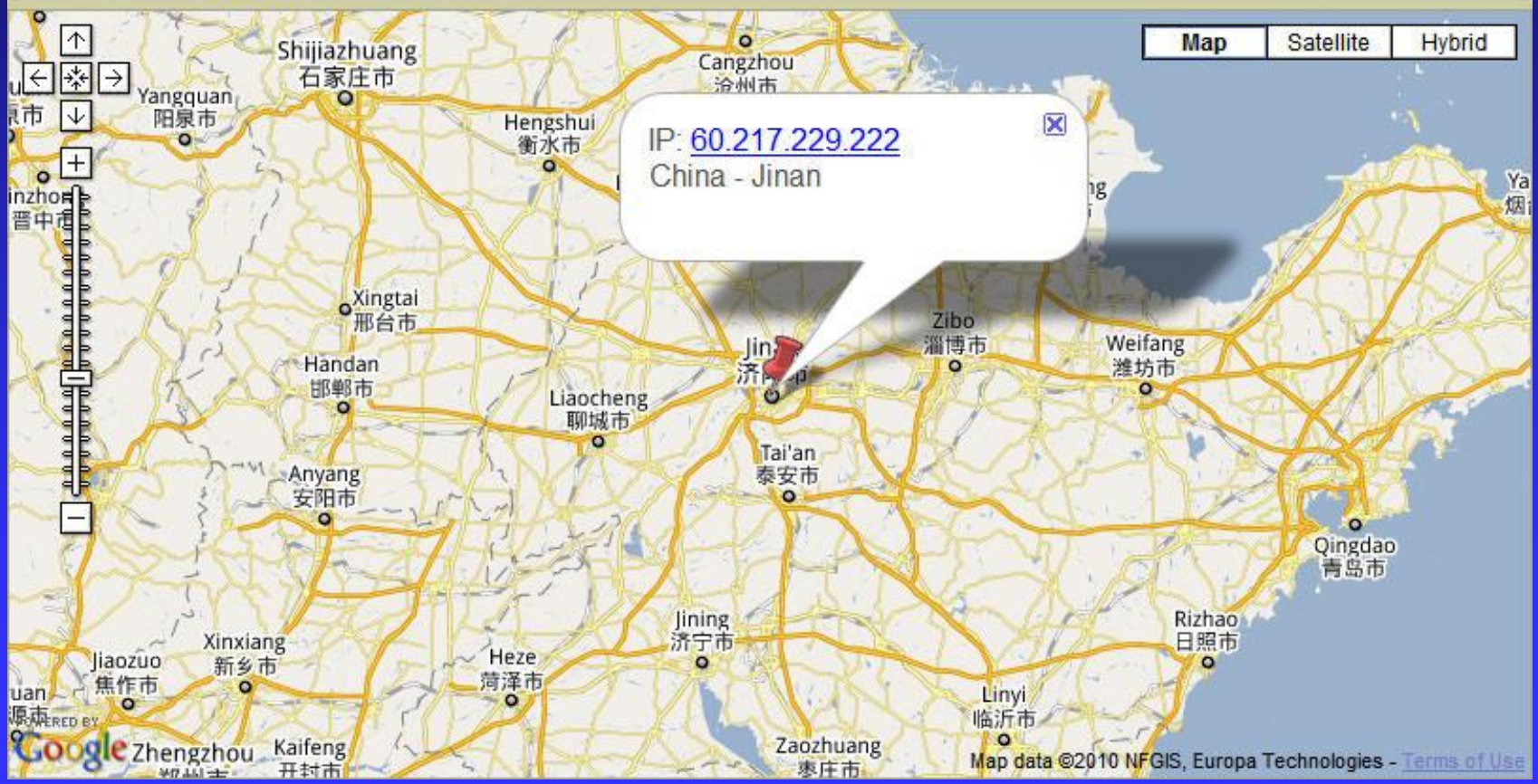
For more info: "*Comm Server IP Programmer's Guide and Ref.*"

Logging by FTP Server Exits – FTP brute force attack

```
AES824I FTP OPEN CONNECTION,IP=60.217.229.222,PORT=53612,TIME=20:56:57.21
AES826I FTP CMD=USER    ,USER=         ,TIME=20:56:57.57,ARG=Administrator
AES826I FTP CMD=USER    ,USER=ADMINIST,TIME=20:56:57.88,ARG=Administrator
AES826I FTP CMD=PASS    ,USER=ADMINIST,TIME=20:56:58.20,ARG=
AES825I FTP LOGIN,USER=ADMINIST,TIME=20:56:58.20
AES826I FTP CMD=PASS    ,USER=         ,TIME=20:56:58.54,ARG=
AES826I FTP CMD=USER    ,USER=         ,TIME=20:56:58.87,ARG=Administrator
AES826I FTP CMD=QUIT    ,USER=ADMINIST,TIME=20:56:59.20,ARG=
AES824I FTP OPEN CONNECTION,IP=60.217.229.222,PORT=55787,TIME=20:56:59.53
AES826I FTP CMD=USER    ,USER=         ,TIME=20:56:59.90,ARG=Administrator
AES826I FTP CMD=USER    ,USER=ADMINIST,TIME=20:57:00.23,ARG=Administrator
AES826I FTP CMD=PASS    ,USER=ADMINIST,TIME=20:57:00.55,ARG=
AES825I FTP LOGIN,USER=ADMINIST,TIME=20:57:00.55
AES826I FTP CMD=PASS    ,USER=         ,TIME=20:57:00.88,ARG=
AES826I FTP CMD=USER    ,USER=         ,TIME=20:57:01.20,ARG=Administrator
AES826I FTP CMD=QUIT    ,USER=ADMINIST,TIME=20:57:01.53,ARG=
AES824I FTP OPEN CONNECTION,IP=60.217.229.222,PORT=58138,TIME=20:57:01.86
AES826I FTP CMD=USER    ,USER=         ,TIME=20:57:02.23,ARG=Administrator
AES826I FTP CMD=USER    ,USER=ADMINIST,TIME=20:57:02.80,ARG=Administrator
AES826I FTP CMD=PASS    ,USER=ADMINIST,TIME=20:57:03.13,ARG=
AES825I FTP LOGIN,USER=ADMINIST,TIME=20:57:03.13
AES826I FTP CMD=PASS    ,USER=         ,TIME=20:57:03.47,ARG=
AES826I FTP CMD=USER    ,USER=         ,TIME=20:57:03.79,ARG=Administrator
AES826I FTP CMD=QUIT    ,USER=ADMINIST,TIME=20:57:03.79,ARG=
AES824I FTP OPEN CONNECTION,IP=60.217.229.222,PORT=60266,TIME=20:57:04.10
```

Who's this guy?

IP: 60.217.229.222 [Ripe.Net] - [DNS] - [Tracert]
Domain: 60.217.229.222 [Whois]
Country: CN - China
State/Region: Shandong
City: Jinan
Integrate your Widget into your website.



Sample FTP Session 1

```
ftp 137.72.43.207
 EZY2640I Using 'TCPIP.FTP.DATA' for local site configuration parameters.
 EZA1450I IBM FTP CS V1R7
 EZA1554I Connecting to:   137.72.43.207 port: 21.
 220-FTPD1 IBM FTP CS V1R7 at OS16.AESCLEVER.COM, 14:51:31 on 2007-02-09.
 220 Connection will not timeout.
 EZA1459I NAME (137.72.43.207:AESDJC1):
p390
 EZA1701I >>> USER p390
 331 Send password please.
 EZA1789I PASSWORD:
EZA1701I >>> PASS
 230 P390 is logged on.  Working directory is "AESDJC1.".
 EZA1460I Command:
bin
 EZA1701I >>> TYPE I
 200 Representation type is Image
 EZA1460I Command:
get 'aesdjc1.xmi' 'aesdjc1.xmi' (replace
 EZA1701I >>> PORT 137,72,43,240,6,139
 200 Port request OK.
 EZA1701I >>> RETR 'aesdjc1.xmi'
 125 Sending data set AESDJC1.XMI FIXrecfm 80
 250 Transfer completed successfully.
 EZA1617I 166400 bytes transferred in 2.180 seconds.  Transfer rate 76.33 Kbytes
    /sec.
```

Sample FTP Session 1 – Logging by FTP Exits

```
FTP OPEN CONNECTION,IP=137.72.43.240,PORT= 1674,TIME=14:51:13.67

FTP CMD=USER     ,USER=         ,TIME=14:51:16.01,ARG=p390

FTP CMD=PASS     ,USER=P390     ,TIME=14:51:17.81,ARG=

FTP LOGIN,USER=P390     ,TIME=14:51:17.81

FTP CMD=TYPE     ,USER=P390     ,TIME=14:51:23.03,ARG=I

FTP CMD=PORT     ,USER=P390     ,TIME=14:51:34.37,ARG=137,72,43,240,6,139

FTP CMD=RETR     ,USER=P390     ,TIME=14:51:34.40,ARG='aesdjc1.xmi'

FTP POST,CMD=RETR,USER=P390
    ,IP=137.72.43.240,TYPE=MVS/SEQ,RC=250,REASON=0,TIME=14:51:36.93
    SESSIONID=FTPD100011, CPU TIME=0.829
```

FTCHKIP

FTCHKCMD

FTCHKCMD

FTCHKPWD

FTCHKCMD

FTCHKCMD

FTCHKCMD

FTPOSTPR

Sample FTP Session 1

How to interpret the PORT command

PORT 137,72,43,240,6,139

IP Address of the client: **137.72.43.240**

Port of the client: **256*6 + 139 = 1675**

Sample FTP Session 1: Active FTP

Sample FTP Session 1 – FTP Server SMF data

```
FTPS:RETR,IP=137.72.43.240,PORT=21/1674,RC=250,
User=P390,Format=S/S/I,ABND=
Start=15:51:34,End=15:51:34,Bytes=166400,
Elapsed=0.010sec,Throughput=16640.00KB/sec
DSN1=AESDJC1.XMI/,DSN2=/
```

Format:

Data set type:	P – partitioned, **S** – sequential, H – HFS
Mode:	**S** – stream, B – block, C – compressed
Data format:	A – ASCII, E – EBCDIC, **I** – image (binary), D – double-byte, U – UCS-2

Sample FTP Session 2

```
 EZA1460I Command:
put 'aesdjc1.xmi' 'aesdjc1.small'
 EZA1701I >>> SITE FIXrecfm 80 LRECL=80 RECFM=FB BLKSIZE=3120
 200 SITE command was accepted
 EZA1701I >>> PORT 137,72,43,240,6,142
 200 Port request OK.
 EZA1701I >>> STOR 'aesdjc1.small'
 125 Storing data set AESDJC1.SMALL
 451-System completion code and reason: D37-04
 451-Data set is out of space.
 451 Transfer aborted due to file error.
 EZA1460I Command:
quit
 EZA1701I >>> QUIT
 221 Quit command received. Goodbye.
 READY
```

Sample FTP Session 2 – logging by FTP Exits

```
FTP CMD=SITE    ,USER=P390    ,TIME=14:53:28.45,ARG=FIXrecfm 80 LRECL=80
   RECFM=FB BLKSIZE=3120

FTP CMD=PORT    ,USER=P390    ,TIME=14:53:28.50,ARG=137,72,43,240,6,142

FTP CMD=STOR    ,USER=P390    ,TIME=14:53:28.52,ARG='aesdjc1.small'

FTP POST,CMD=STOR,USER=P390
   ,IP=137.72.43.240,TYPE=MVS/SEQ,RC=451,REASON=4,TIME=14:53:29.61

FTP REPLY=Transfer aborted due to file error.

FTP CMD=QUIT    ,USER=P390    ,TIME=14:53:31.48,ARG=
```

Sample FTP Session 2 – FTP Server SMF Record

```
FTPS:STOR,IP=137.72.43.240,PORT=21/1674,RC=451,
User=P390,Format=S/S/I,ABND=
Start=15:53:28,End=15:53:29,Bytes=166400,
Elapsed=0.500sec,Throughput=332.80KB/sec
DSN1=AESDJC1.SMALL/,DSN2=/
```

Reply Code 451: Requested action aborted. Local error in processing.

SMF 119 TCP Connection Termination and FTP Server Record

SysPoint | Connect Expert | StackView | LinkView | Critical Resources | PinPoint

TCP SMF — March 2, 2009 4:10:17 PM

Refresh

6 items found, displaying all items.1

Count	Date	Time	Rec Type	Address	Foreign Port	Local Port	Job Name	Job Id	Byte In	Byte Out	Segment In	Segment Out	Session RTT
1	03/02/2009	17:57:23:75	Term	137.72.43.142	4919	21	TCPIP	TCPIP	100	393	16	12	90
2	03/02/2009	17:57:22:15	Term	137.72.43.142	4922	20	TCPIP	TCPIP	0	2460	4	5	3
3	03/02/2009	17:57:21:93	Init	137.72.43.142	4922	20	TCPIP	TCPIP	0	0	0	0	0
4	03/02/2009	17:57:08:16	Init	137.72.43.142	4919	21	TCPIP	TCPIP	0	0	0	0	0
5	03/02/2009	17:57:06:99	Term	137.72.43.142	4546	21	TCPIP	TCPIP	35	233	10	8	62
6	03/02/2009	17:19:06:36	Init	137.72.43.142	4546	21	TCPIP	TCPIP	0	0	0	0	0

Export options: CSV | Excel | XML | PDF

FTP SMF Log — March 2, 2009 4:11:34 PM

Refresh

FTP Server

Count	Date MM/dd/yyyy	Start Time	End Time	Address	Foreign Port	User ID	Data Mode	Data Format	Data Set Type	Total Bytes	Return Code	Description	Data Set Name	Member	Abnormal End
1	03/02/2009	17:57:22:10	17:57:22:11	137.72.43.207	4919	AESDJC2	Stream	ASCII	Partitioned	2,460	250	Requested file action okay; completed.	AESDJC1.MAIN.CNTL	ASM	-
2	03/02/2009	17:47:26:15	17:47:26:15	137.72.43.207	3186	AESQMS	Stream	ASCII	Sequential	651	250	Requested file action okay; completed.	AESQMS.AES3U.LOG	-	-
3	03/02/2009	17:47:25:76	17:47:25:76	137.72.43.207	3188	AESQMS	Stream	I	Sequential	1,309	250	Requested file action okay; completed.	AESQMS.F2DEFS.TXT	-	-

TCP Session RTT Report

CleverView(TM) for TCP/IP BatchPR

APPLIED EXPERT SYSTEMS, INC.

System: OS1X Sysplex: ADCDPL TCP/IP Stack: TCPIP

Date: 10/14/2009 (2009.287)
Shift: 4, From: 18:00 To: 20:00

LOCATION	Remote IP Addr	Appl	# Sessions	Round Trip Time (ms) Avg	Std Dev	Min	Max	Segments In	Segments Out
AES2	137.72.43.239								
		AESTCP80	237	3.09	15.119	0	174	475	741
		FTPD1	87	26.64	56.467	1	275	129	257
		HTTPD1	79	1.85	6.522	0	59	130	238
		NPMTCPI5	16	0.50	0.500	0	1	40	56
		TCPIP	87	5.00	32.140	0	301	94	217
			506	7.19	30.318	0	301	868	1509
HOST16	137.72.43.252								
		AESTCP80	79	9.28	44.817	3	404	158	237
		TCPIP	14	5.29	4.131	3	20	28	42
			93	8.68	41.362	3	404	186	279
SUBNET1	137.72.43.*								
		AESTCP80	316	4.64	26.091	0	404	633	978
		FTPD1	**87**	**26.64**	**56.467**	**1**	**275**	**129**	**257**
		HTTPD1	79	1.85	6.522	0	59	130	238
		NPMTCPI5	16	0.50	0.500	0	1	40	56
		TCPIP	103	5.49	29.919	0	301	3378	3964
			601	7.49	32.281	0	404	4310	5493

# Appl	Sessions	Round Trip Time (ms) Avg	Std Dev	Min	Max	Segments In	Segments Out
AESTCP80	316	4.64	26.091	0	404	633	978
FTPD1	**87**	**26.64**	**56.467**	**1**	**275**	**129**	**257**
HTTPD1	79	1.85	6.522	0	59	130	238
NPMTCPI5	16	0.50	0.500	0	1	40	56
TCPIP	103	5.49	29.896	0	301	3378	3964

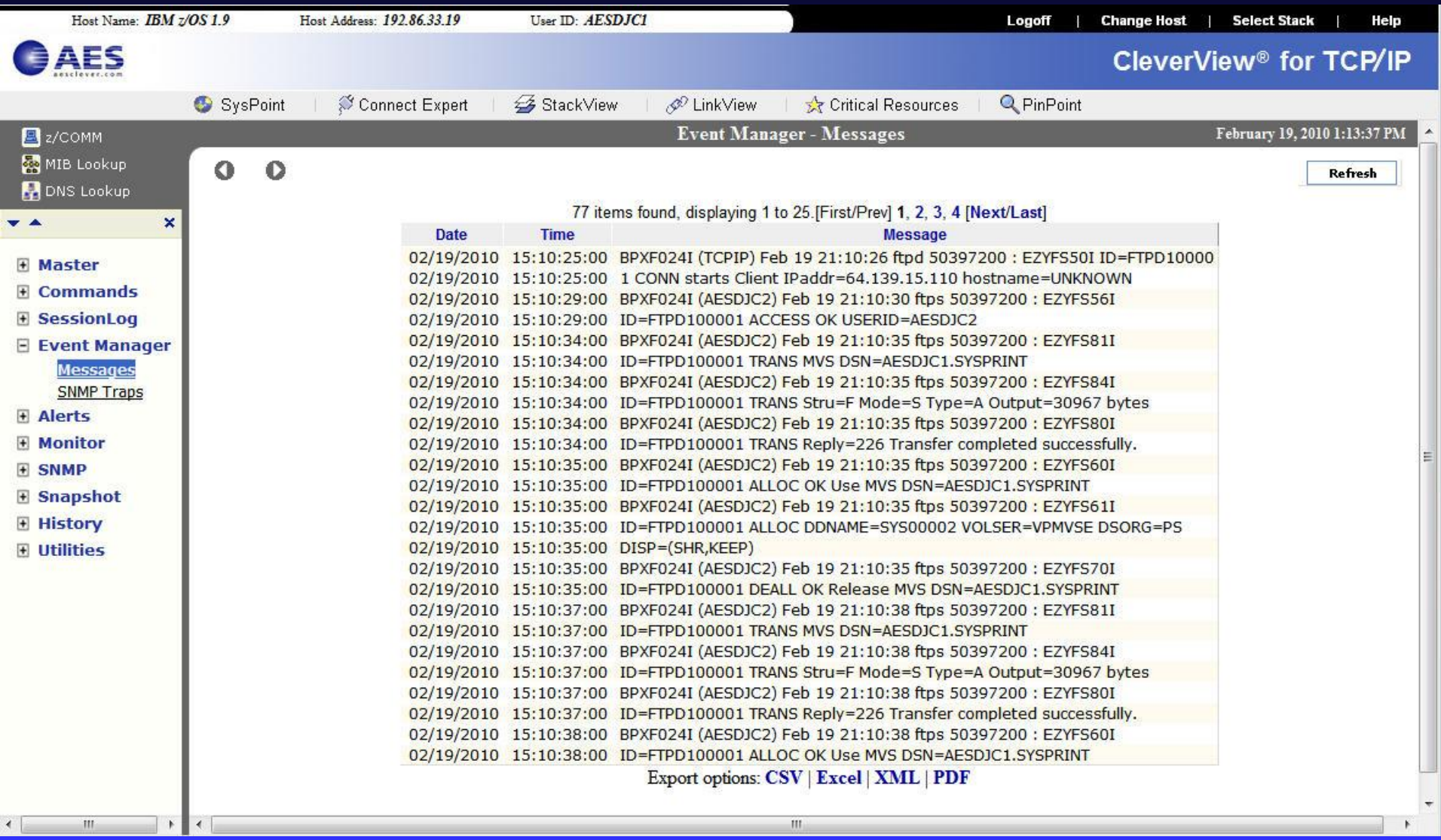
FTP Server Logging

- FTP Server can log activities to SyslogD via the following FTP.DATA options:
  - FTPLOGGING TRUE | TRUENODNS
  - ANONYMOUSFTPLOGGING TRUE
- Nine events are logged:
  - CONN connectivity
  - SECURE security (TLS/SSL, Kerberos)
  - ACCESS login
  - ALLOC file and data set allocation
  - DEALL file and data set de-allocation
  - TRANS file transfer
  - SUBMIT JES job submission
  - QUERY SQL query
  - ABEND abnormal termination
- Each activity logging message has a message number within the range of EZYFS50 to EZYFS95 – prefixed by BPFX024I

FTPLOGGING output – centralized BPXF024I messages - log

```
---- FRIDAY, 19 FEB 2010 ----
15:10:25 BPXF024I (TCPIP) Feb 19 21:10:26 ftpd 50397200 : EZYFS50I ID=FTPD10000
15:10:25 1 CONN   starts Client IPaddr=137.72.43.32 hostname=UNKNOWN
15:10:29 BPXF024I (AESDJC2) Feb 19 21:10:30 ftps 50397200 : EZYFS56I
15:10:29 ID=FTPD100001 ACCESS OK     USERID=AESDJC2
15:10:35 BPXF024I (AESDJC2) Feb 19 21:10:35 ftps 50397200 : EZYFS60I
15:10:35 ID=FTPD100001 ALLOC  OK     Use MVS DSN=AESDJC1.SYSPRINT
15:10:35 BPXF024I (AESDJC2) Feb 19 21:10:35 ftps 50397200 : EZYFS61I
15:10:35 ID=FTPD100001 ALLOC  DDNAME=SYS00002 VOLSER=VPMVSE DSORG=PS
15:10:35 DISP=(SHR,KEEP)
15:10:35 BPXF024I (AESDJC2) Feb 19 21:10:35 ftps 50397200 : EZYFS70I
15:10:35 ID=FTPD100001 DEALL  OK     Release MVS DSN=AESDJC1.SYSPRINT
15:10:34 BPXF024I (AESDJC2) Feb 19 21:10:35 ftps 50397200 : EZYFS81I
15:10:34 ID=FTPD100001 TRANS  MVS DSN=AESDJC1.SYSPRINT
15:10:34 BPXF024I (AESDJC2) Feb 19 21:10:35 ftps 50397200 : EZYFS84I
15:10:34 ID=FTPD100001 TRANS  Stru=F Mode=S Type=A Output=30967 bytes
15:10:34 BPXF024I (AESDJC2) Feb 19 21:10:35 ftps 50397200 : EZYFS80I
15:10:34 ID=FTPD100001 TRANS  Reply=226 Transfer completed successfully.
15:10:38 BPXF024I (AESDJC2) Feb 19 21:10:38 ftps 50397200 : EZYFS60I
15:10:38 ID=FTPD100001 ALLOC  OK     Use MVS DSN=AESDJC1.SYSPRINT
15:10:38 BPXF024I (AESDJC2) Feb 19 21:10:38 ftps 50397200 : EZYFS61I
15:10:38 ID=FTPD100001 ALLOC  DDNAME=SYS00003 VOLSER=VPMVSE DSORG=PS
15:10:38 DISP=(SHR,KEEP)
15:10:38 BPXF024I (AESDJC2) Feb 19 21:10:38 ftps 50397200 : EZYFS70I
15:10:38 ID=FTPD100001 DEALL  OK     Release MVS DSN=AESDJC1.SYSPRINT
15:10:37 BPXF024I (AESDJC2) Feb 19 21:10:38 ftps 50397200 : EZYFS81I
15:10:37 ID=FTPD100001 TRANS  MVS DSN=AESDJC1.SYSPRINT
15:10:37 BPXF024I (AESDJC2) Feb 19 21:10:38 ftps 50397200 : EZYFS84I
15:10:37 ID=FTPD100001 TRANS  Stru=F Mode=S Type=A Output=30967 bytes
15:10:37 BPXF024I (AESDJC2) Feb 19 21:10:38 ftps 50397200 : EZYFS80I
15:10:37 ID=FTPD100001 TRANS  Reply=226 Transfer completed successfully.
```

FTPLOGGING output – centralized BPXF024I messages - GUI



FTP Server Tracing

- TRACE and DEBUG statements in FTP.DATA
  - TRACE is equivalent to DEBUG BAS, which includes:
    - DEBUG CMD - command
    - DEBUG INT – init and term of FTP session
    - DEBUG FSC – details of APPE, STOR, STOU, RETR, DELE, RNFR, RNTO
    - DEBUG SOC – interface between FTP and network

- Use the SITE command to turn on tracing dynamically only for the duration of an FTP session
  - Requires: DEBUGONSITE TRUE be specified in FTP.DATA
  - z/OS example : **site debug=bas**
  - MS/DOS example: **quote site debug=bas**

- Use the MODIFY command
  - F jobname,DEBUG=(BAS)
  - F jobname,DEBUG=(NONE)

- Output in SYSLOG

FTP Server Tracing Output

```
13:40:24.51 AESDJC1          00000290  F FTPD1,DEBUG=(BAS)
13:40:24.56 STC05692 00000090          BPXF024I (TCPIP) Feb 19 19:40:24 ftpd 65548 : EZYFT82I ACTIVE SERVER 197
                        197 00000090   TRACES - CMD INT FSC(1) SOC(1)
13:40:24.56 STC05692 00000090          +EZYFT82I ACTIVE SERVER TRACES - CMD INT FSC(1) SOC(1)
. . . . . .
                        203 00000090   ket: new session for 137.72.43.32 port 60265
13:40:44.91 STC05692 00000090          BPXF024I (TCPIP) Feb 19 19:40:44 ftpd 65548 : SD1152 spawn_ftps: 204
                        204 00000090   entered
13:40:44.92 STC05692 00000090          BPXF024I (TCPIP) Feb 19 19:40:44 ftpd 65548 : SD0359 accept_client: 205
                        205 00000090   prepare to accept another client
. . . . . .
                        208 00000090   tmpmsg is 220-FTPD1 IBM FTP CS V1R9 at S0W1, 19:40:44 on 2010-02-19.
13:40:44.99 STC06087 00000090          BPXF024I (TCPIP) Feb 19 19:40:44 ftpd 83951627 : SD1307 spawn_ftps: 209
                        209 00000090   tmpmsg is 220 Connection will close if idle for more than 50 minutes.
13:40:44.99 STC06087 00000090          BPXF024I (TCPIP) Feb 19 19:40:44 ftpd 83951627 : EZYFS50I ID=FTPD10002
                                       210
                        210 00000090   0 CONN   starts Client IPaddr=137.72.43.32 hostname=UNKNOWN
. . . . . .
13:40:45.18 STC06087 00000090          BPXF024I (TCPIP) Feb 19 19:40:45 ftps 83951627 : GU1125 chkVerRel: 214
                        214 00000090   system information for S0W1: z/OS version 1 release 9 (2097)
13:40:45.18 STC06087 00000090          BPXF024I (TCPIP) Feb 19 19:40:45 ftps 83951627 : PR0308 parse_cmd: 215
                        215 00000090   entered
13:40:46.73 STC06087 00000090          BPXF024I (TCPIP) Feb 19 19:40:46 ftps 83951627 : PR0487 parse_cmd: 216
                        216 00000090   >>> USER aesdjc1
13:40:46.73 STC06087 00000090          BPXF024I (TCPIP) Feb 19 19:40:46 ftps 83951627 : SR3136 reply: --> 217
                        217 00000090   331 Send password please.
13:40:48.41 STC06087 00000090          BPXF024I (TCPIP) Feb 19 19:40:48 ftps 83951627 : PR0492 parse_cmd: 218
                        218 00000090   >>> PASS ******
13:40:48.42 STC06087 00000090          BPXF024I (TCPIP) Feb 19 19:40:48 ftps 83951627 : RA0627 RACF_MIXED_CAS
                                       219
                        219 00000090   E_PASSWORDS_ENABLED 0
```

CTRACE – Packet Tracing (SYSTCPDA)

- Set up External Writer Proc

  E.g., SYS1.PROCLIB(AESWRT):

```
//IEFPROC EXEC PGM=ITTTRCWR,REGION=0K,TIME=1440,DPRTY=15
//TRCOUT01 DD DISP=SHR,DSN=trace.dataset
```

- Set up tracing parameters

  E.g., SYS1.PARMLIB(CTAESPRM):

```
TRACEOPTS ON WTR(AESWRT)
```

CTRACE – Packet Tracing (SYSTCPDA)

- To Start Tracing:
  - TRACE CT,WTRSTART=**AESWRT**
  - V TCPIP,,PKT,CLEAR
  - V TCPIP,,PKT,LINKN=ETH1,ON,FULL,PROT=TCP,IP=<ip addr>
  - TRACE CT,ON,COMP=SYSTCPDA,SUB=(TCPIP),PARM=**CTAESPRM**
- To View Tracing Status:
  - D TRACE,WTR=**AESWRT**
    - Verify that the external writer is active
  - D TCPIP,,NETSTAT,DE
    - Verify that **TrRecCnt** is non-zero and incrementing
- To Stop Tracing:
  - V TCPIP,,PKT,OFF
  - TRACE CT,OFF,COMP=SYSTCPDA,SUB=(TCPIP)
  - TRACE CT,WTRSTOP=**AESWRT**,FLUSH

CTRACE – OSAENTA Tracing (SYSTCPOT)

- Trace packets to a host attached to an OSA-Express2.
- The host can be an LPAR with **z/OS, z/VM** or **Linux**.
- The trace function is controlled by z/OS Communication Server, while the data is collected in the OSA at the network port.

- Pre-Reqs:
  - Install the required PTFs for z/OS V1R8 (APAR PK36947).
  - Install the microcode for the OSA (2094DEVICE PSP and the 2096DEVICE PSP).
  - Update the OSA using the Hardware Management Console (HMC) to:

    Define more data devices to systems that will use the trace function.

    Set the security for the OSA:

    LOGICAL PARTITION - Only packets from the LPAR

    CHPID - All packets using this CHPID
  - Verify the TRLE definitions for the OSA that it has one DATAPATH address available for tracing. Note that **two** DATAPATH addresses are required – one for data transfers and the other for trace data.

CTRACE – OSAENTA Tracing (SYSTCPOT)

- To Start Tracing:

```
TRACE CT,WTRSTART=AESWRT
V TCPIP,,OSAENTA,PORTNAME=<port>,CLEAR
V TCPIP,,OSAENTA,PORTNAME=<port>,ON,NOFILTER=ALL
TRACE CT,ON,COMP=SYSTCPOT,SUB=(TCPIP),PARM=CTAESPRM
```

- To Stop Tracing:

```
V TCPIP,,OSAENTA,PORTNAME=<port>,OFF
TRACE CT,OFF,COMP=SYSTCPOT,SUB=(TCPIP)
TRACE CT,WTRSTOP=AESWRT,FLUSH
```

- To View Tracing Status:

```
D TCPIP,,NETSTAT,DEVLINKS
D TRACE,WTR=AESWRT
```

CTRACE Packet Trace Decoding – IPCS JCL

```
//TSO    EXEC PGM=IKJEFT01,DYNAMNBR=60,
// PARM='%BLSCDDIR DSNAME(&SYSUID..BATCH.DDIR)
  VOLUME(AES003)'
//SYSPROC  DD DISP=SHR,DSN=SYS1.SBLSCLI0
//TRACE    DD DISP=SHR,DSN=trace.dataset    <=== INPUT
//IPCSPRNT DD SYSOUT=*
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD *
  IPCS NOPARM
    DROPD FILE(TRACE)
    SETDEF NOCONFIRM PRINT NOTERM
    CTRACE DDNAME(TRACE) COMP(SYSTCPDA) +
        SUB((TCPIP)) OPTIONS(( FTP(20,21) )) FULL GMT
  END /* IPCS */
//
```

Specify COMP(SYSTCPOT) for OSAENTA trace

CTRACE Packet Trace Decoding – IPCS Output

```
IPCS PRINT LOG FOR USER AESDJC1                                                    05:15:13 02/24/08
_____________________________________________________________________________________________
 COMPONENT TRACE FULL FORMAT
 SYSNAME(ADCD)
 COMP(SYSTCPDA)SUBNAME((TCPIP))
 OPTIONS((FTP(20,21)))
 z/OS TCP/IP Packet Trace Formatter, (C) IBM 2000-2005, 2005.047
 FILE(TRACE)
**** 2008/02/22
RcdNr Sysname  Mnemonic Entry Id   Time Stamp    Description
----- -------- -------- -------- --------------- --------------------------------
--------------------------------------------------------------------------------
804059 ADCD     PACKET   00000004 20:48:42.883175 Packet Trace
 From Interface    : ETH1             Device: LCS Ethernet     Full=52
  Tod Clock        : 2008/02/22 20:48:42.883162                Intfx: 4
  Sequence #       : 0                Flags: Pkt
 IpHeader: Version : 4                Header Length: 20
  Tos              : 00               QOS: Routine Normal Service
  Packet Length    : 52               ID Number: AD04
  Fragment         : DontFragment     Offset: 0
  TTL              : 64               Protocol: TCP            CheckSum: 23F2 FFFF
  Source           : 137.72.43.110
  Destination      : 137.72.43.207
 TCP
  Source Port      : 28265 ()         Destination Port: 21    (ftp)
  Sequence Number  : 1439084340       Ack Number: 0
  Header Length    : 32               Flags: Syn
  Window Size      : 65534            CheckSum: 91D2 FFFF Urgent Data Pointer: 0000
   Option          : Max Seg Size Len: 4 MSS: 1460
   Option          : NOP
   Option          : Window Scale OPT Len: 3 Shift: 0
   Option          : NOP
   Option          : NOP
   Option          : SACK Permitted
IP Header          : 20
000000 45000034 AD044000 400623F2 89482B6E  89482BCF
```

z/VM Packet Trace

- To enable the trace:
  - NETSTAT OBEY PACKETTRACESIZE 256
  - NETSTAT OBEY TRACEONLY ETH0 ENDTRACEONLY
- To start data collection:
  - TRSOURCE ID TCP TYPE GT BLOCK FOR USER tcpip_userid
  - TRSOURCE ENABLE ID TCP
- To stop data collection:
  - NETSTAT OBEY PACKETTRACESIZE 0
  - NETSTAT OBEY TRACEONLY ENDTRACEONLY
  - TRSOURCE DISABLE ID TCP
- To analyze a TRF trace file:
  - IPFORMAT command
  - Use the TRF2TCPD utility to convert the TRF file to pcap (tcpdump) format

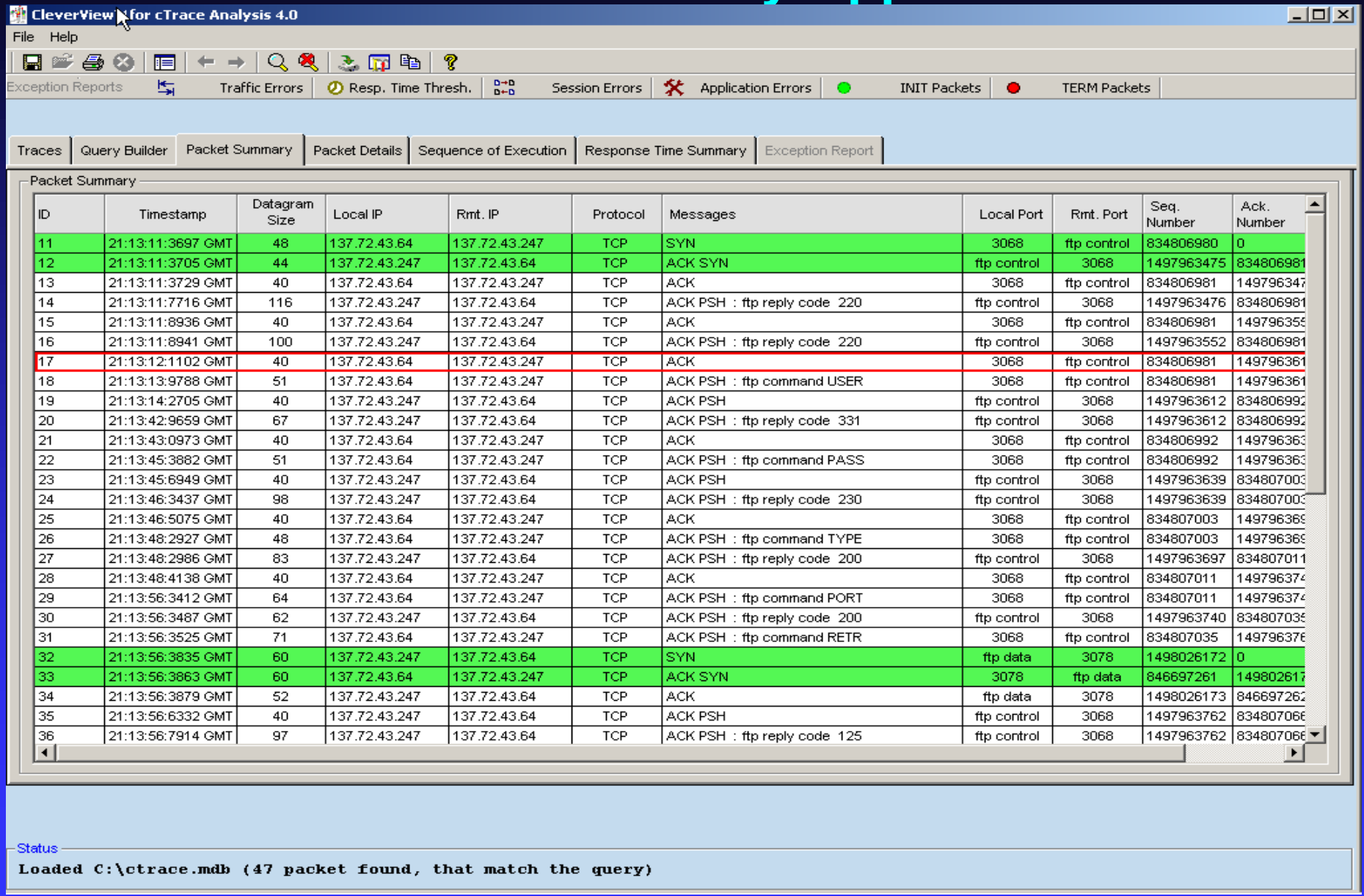
Packet Trace Analysis

- Analyze one FTP session at a time
- Separate the Control Session from the Data Session
- Check session initiation and termination
- Check FTP commands and replies
- Look for packet retransmissions and unusual long response times
- Elapsed time between packets
- TCP window size

Packet Trace Analysis

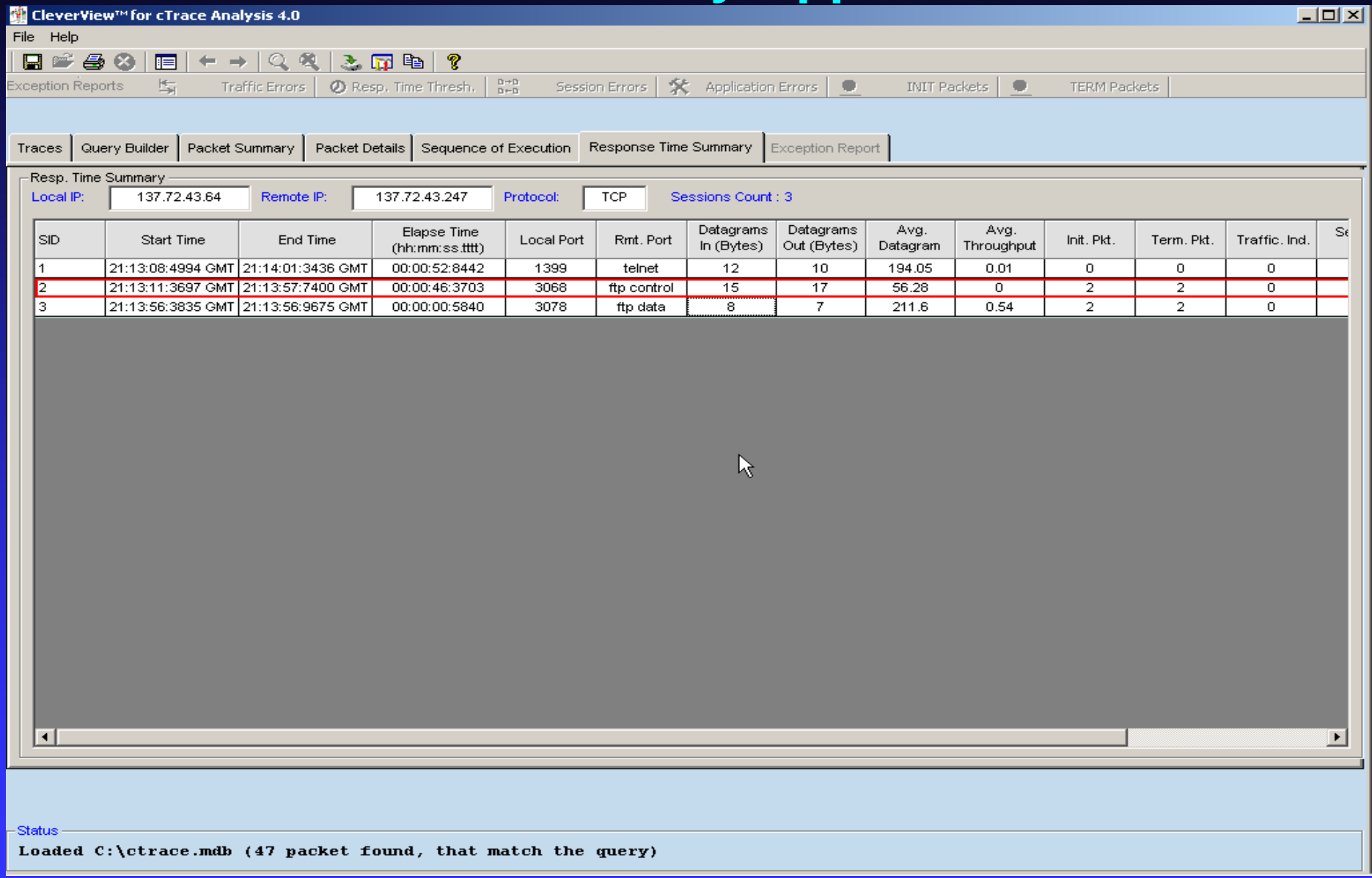
- RST flag in TCP header
  - Abnormal condition and the receiver wants to abort the connection; e.g.,
    - Receipt of any TCP segment from a host with which the receiver does not currently have a connection
    - Receipt of an invalid/incorrect Sequence Number or Acknowledge Number
    - Receipt of a SYN on a port without a listener
  - Possible time-out on the Control Connection
  - Issue netstat command on both sides to verify connections
- Run packet traces on both sides and compare
- Capture “good” traces for future comparisons

Packet Trace – Unfiltered by Application Ports



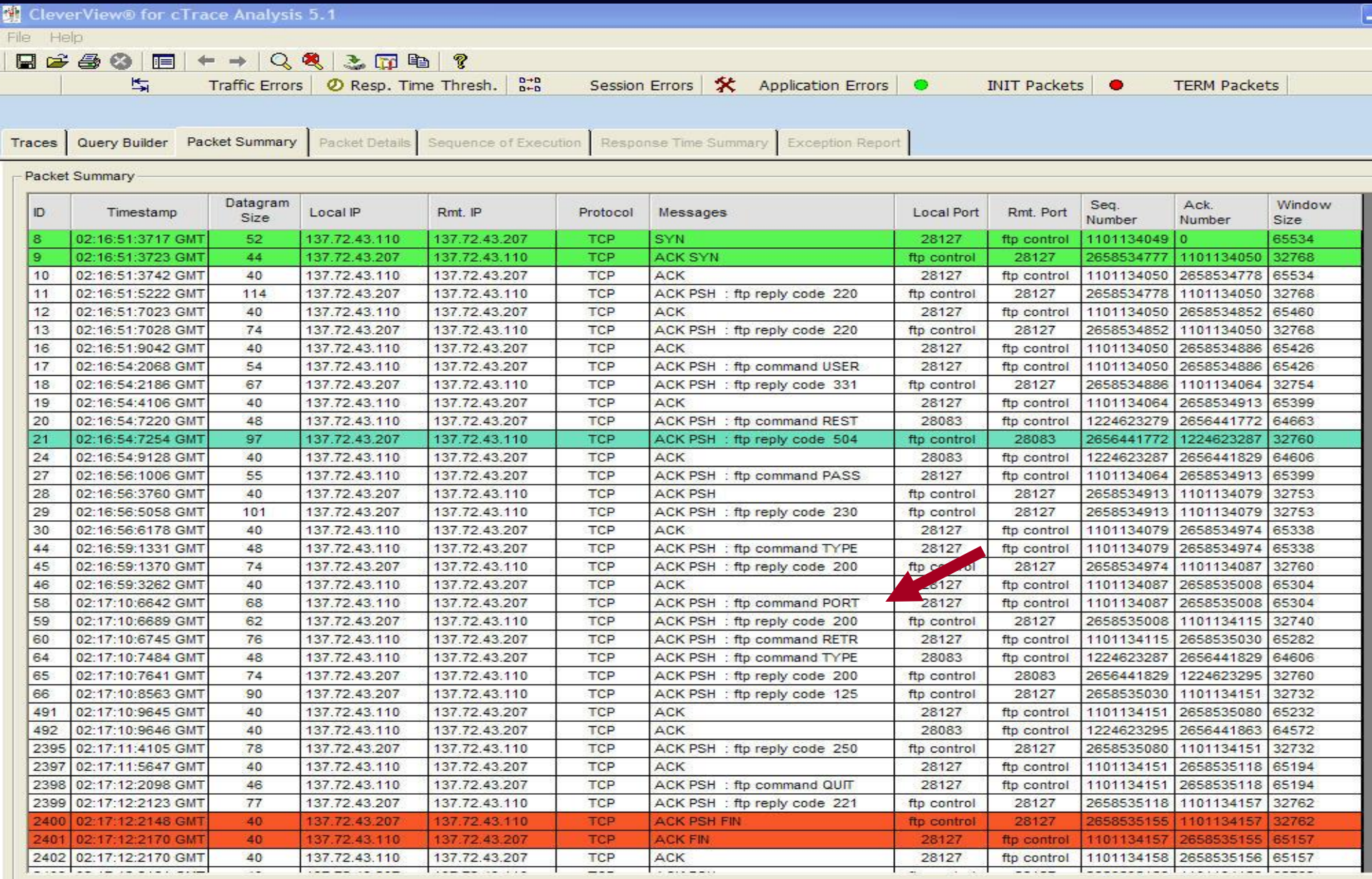
ID	Timestamp	Datagram Size	Local IP	Rmt. IP	Protocol	Messages	Local Port	Rmt. Port	Seq. Number	Ack. Number
11	21:13:11:3697 GMT	48	137.72.43.64	137.72.43.247	TCP	SYN	3068	ftp control	834806980	0
12	21:13:11:3705 GMT	44	137.72.43.247	137.72.43.64	TCP	ACK SYN	ftp control	3068	1497963475	834806981
13	21:13:11:3729 GMT	40	137.72.43.64	137.72.43.247	TCP	ACK	3068	ftp control	834806981	149796347
14	21:13:11:7716 GMT	116	137.72.43.247	137.72.43.64	TCP	ACK PSH : ftp reply code 220	ftp control	3068	1497963476	834806981
15	21:13:11:8936 GMT	40	137.72.43.64	137.72.43.247	TCP	ACK	3068	ftp control	834806981	149796359
16	21:13:11:8941 GMT	100	137.72.43.247	137.72.43.64	TCP	ACK PSH : ftp reply code 220	ftp control	3068	1497963552	834806981
17	21:13:12:1102 GMT	40	137.72.43.64	137.72.43.247	TCP	ACK	3068	ftp control	834806981	149796361
18	21:13:13:9788 GMT	51	137.72.43.64	137.72.43.247	TCP	ACK PSH : ftp command USER	3068	ftp control	834806981	149796361
19	21:13:14:2705 GMT	40	137.72.43.247	137.72.43.64	TCP	ACK PSH	ftp control	3068	1497963612	834806992
20	21:13:42:9659 GMT	67	137.72.43.247	137.72.43.64	TCP	ACK PSH : ftp reply code 331	ftp control	3068	1497963612	834806992
21	21:13:43:0973 GMT	40	137.72.43.64	137.72.43.247	TCP	ACK	3068	ftp control	834806992	149796363
22	21:13:45:3882 GMT	51	137.72.43.64	137.72.43.247	TCP	ACK PSH : ftp command PASS	3068	ftp control	834806992	149796363
23	21:13:45:6949 GMT	40	137.72.43.247	137.72.43.64	TCP	ACK PSH	ftp control	3068	1497963639	834807003
24	21:13:46:3437 GMT	98	137.72.43.247	137.72.43.64	TCP	ACK PSH : ftp reply code 230	ftp control	3068	1497963639	834807003
25	21:13:46:5075 GMT	40	137.72.43.64	137.72.43.247	TCP	ACK	3068	ftp control	834807003	149796369
26	21:13:48:2927 GMT	48	137.72.43.64	137.72.43.247	TCP	ACK PSH : ftp command TYPE	3068	ftp control	834807003	149796369
27	21:13:48:2986 GMT	83	137.72.43.247	137.72.43.64	TCP	ACK PSH : ftp reply code 200	ftp control	3068	1497963697	834807011
28	21:13:48:4138 GMT	40	137.72.43.64	137.72.43.247	TCP	ACK	3068	ftp control	834807011	149796374
29	21:13:56:3412 GMT	64	137.72.43.64	137.72.43.247	TCP	ACK PSH : ftp command PORT	3068	ftp control	834807011	149796374
30	21:13:56:3487 GMT	62	137.72.43.247	137.72.43.64	TCP	ACK PSH : ftp reply code 200	ftp control	3068	1497963740	834807035
31	21:13:56:3525 GMT	71	137.72.43.64	137.72.43.247	TCP	ACK PSH : ftp command RETR	3068	ftp control	834807035	149796376
32	21:13:56:3835 GMT	60	137.72.43.247	137.72.43.64	TCP	SYN	ftp data	3078	1498026172	0
33	21:13:56:3863 GMT	60	137.72.43.64	137.72.43.247	TCP	ACK SYN	3078	ftp data	846697261	149802617
34	21:13:56:3879 GMT	52	137.72.43.247	137.72.43.64	TCP	ACK	ftp data	3078	1498026173	846697262
35	21:13:56:6332 GMT	40	137.72.43.247	137.72.43.64	TCP	ACK PSH	ftp control	3068	1497963762	834807066
36	21:13:56:7914 GMT	97	137.72.43.247	137.72.43.64	TCP	ACK PSH : ftp reply code 125	ftp control	3068	1497963762	834807066

Packet Trace – Filtered by Application Ports



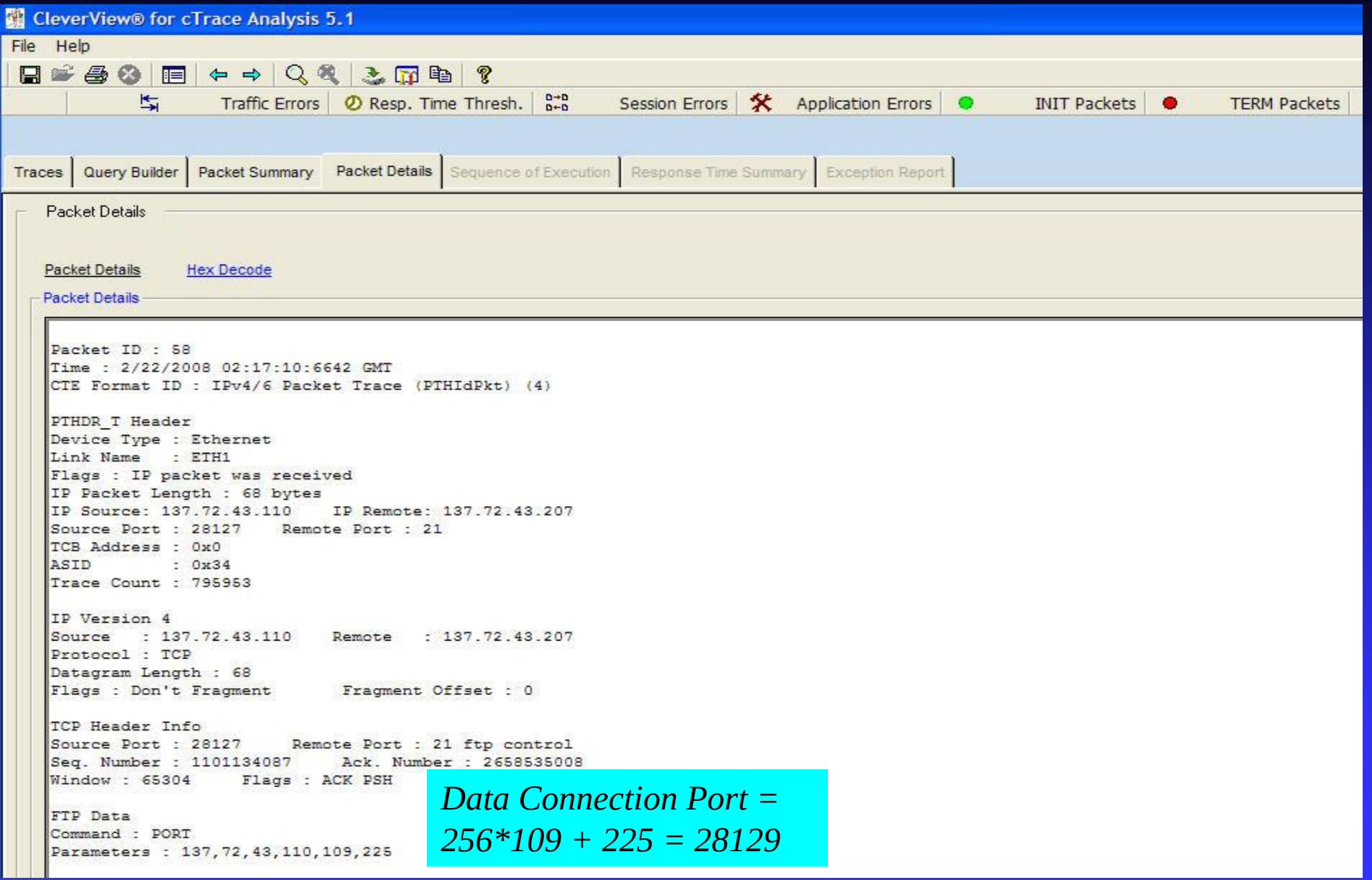
SID	Start Time	End Time	Elapse Time (hh:mm:ss.tttt)	Local Port	Rmt. Port	Datagrams In (Bytes)	Datagrams Out (Bytes)	Avg. Datagram	Avg. Throughput	Init. Pkt.	Term. Pkt.	Traffic. Ind.
1	21:13:08:4994 GMT	21:14:01:3436 GMT	00:00:52:8442	1399	telnet	12	10	194.05	0.01	0	0	0
2	21:13:11:3697 GMT	21:13:57:7400 GMT	00:00:46:3703	3068	ftp control	15	17	56.28	0	2	2	0
3	21:13:56:3835 GMT	21:13:56:9675 GMT	00:00:00:5840	3078	ftp data	8	7	211.6	0.54	2	2	0

Control Connection (Active FTP)



ID	Timestamp	Datagram Size	Local IP	Rmt. IP	Protocol	Messages	Local Port	Rmt. Port	Seq. Number	Ack. Number	Window Size
8	02:16:51:3717 GMT	52	137.72.43.110	137.72.43.207	TCP	SYN	28127	ftp control	1101134049	0	65534
9	02:16:51:3723 GMT	44	137.72.43.207	137.72.43.110	TCP	ACK SYN	ftp control	28127	2658534777	1101134050	32768
10	02:16:51:3742 GMT	40	137.72.43.110	137.72.43.207	TCP	ACK	28127	ftp control	1101134050	2658534778	65534
11	02:16:51:5222 GMT	114	137.72.43.207	137.72.43.110	TCP	ACK PSH : ftp reply code 220	ftp control	28127	2658534778	1101134050	32768
12	02:16:51:7023 GMT	40	137.72.43.110	137.72.43.207	TCP	ACK	28127	ftp control	1101134050	2658534852	65460
13	02:16:51:7028 GMT	74	137.72.43.207	137.72.43.110	TCP	ACK PSH : ftp reply code 220	ftp control	28127	2658534852	1101134050	32768
16	02:16:51:9042 GMT	40	137.72.43.110	137.72.43.207	TCP	ACK	28127	ftp control	1101134050	2658534886	65426
17	02:16:54:2068 GMT	54	137.72.43.110	137.72.43.207	TCP	ACK PSH : ftp command USER	28127	ftp control	1101134050	2658534886	65426
18	02:16:54:2186 GMT	67	137.72.43.207	137.72.43.110	TCP	ACK PSH : ftp reply code 331	ftp control	28127	2658534886	1101134064	32754
19	02:16:54:4106 GMT	40	137.72.43.110	137.72.43.207	TCP	ACK	28127	ftp control	1101134064	2658534913	65399
20	02:16:54:7220 GMT	48	137.72.43.110	137.72.43.207	TCP	ACK PSH : ftp command REST	28083	ftp control	1224623279	2656441772	64663
21	02:16:54:7254 GMT	97	137.72.43.207	137.72.43.110	TCP	ACK PSH : ftp reply code 504	ftp control	28083	2656441772	1224623287	32760
24	02:16:54:9128 GMT	40	137.72.43.110	137.72.43.207	TCP	ACK	28083	ftp control	1224623287	2656441829	64606
27	02:16:56:1006 GMT	55	137.72.43.110	137.72.43.207	TCP	ACK PSH : ftp command PASS	28127	ftp control	1101134064	2658534913	65399
28	02:16:56:3760 GMT	40	137.72.43.207	137.72.43.110	TCP	ACK PSH	ftp control	28127	2658534913	1101134079	32753
29	02:16:56:5058 GMT	101	137.72.43.207	137.72.43.110	TCP	ACK PSH : ftp reply code 230	ftp control	28127	2658534913	1101134079	32753
30	02:16:56:6178 GMT	40	137.72.43.110	137.72.43.207	TCP	ACK	28127	ftp control	1101134079	2658534974	65338
44	02:16:59:1331 GMT	48	137.72.43.110	137.72.43.207	TCP	ACK PSH : ftp command TYPE	28127	ftp control	1101134079	2658534974	65338
45	02:16:59:1370 GMT	74	137.72.43.207	137.72.43.110	TCP	ACK PSH : ftp reply code 200	ftp control	28127	2658534974	1101134087	32760
46	02:16:59:3262 GMT	40	137.72.43.110	137.72.43.207	TCP	ACK	28127	ftp control	1101134087	2658535008	65304
58	02:17:10:6642 GMT	68	137.72.43.110	137.72.43.207	TCP	ACK PSH : ftp command PORT	28127	ftp control	1101134087	2658535008	65304
59	02:17:10:6689 GMT	62	137.72.43.207	137.72.43.110	TCP	ACK PSH : ftp reply code 200	ftp control	28127	2658535008	1101134115	32740
60	02:17:10:6745 GMT	76	137.72.43.110	137.72.43.207	TCP	ACK PSH : ftp command RETR	28127	ftp control	1101134115	2658535030	65282
64	02:17:10:7484 GMT	48	137.72.43.110	137.72.43.207	TCP	ACK PSH : ftp command TYPE	28083	ftp control	1224623287	2656441829	64606
65	02:17:10:7641 GMT	74	137.72.43.207	137.72.43.110	TCP	ACK PSH : ftp reply code 200	ftp control	28083	2656441829	1224623295	32760
66	02:17:10:8563 GMT	90	137.72.43.207	137.72.43.110	TCP	ACK PSH : ftp reply code 125	ftp control	28127	2658535030	1101134151	32732
491	02:17:10:9645 GMT	40	137.72.43.110	137.72.43.207	TCP	ACK	28127	ftp control	1101134151	2658535080	65232
492	02:17:10:9646 GMT	40	137.72.43.110	137.72.43.207	TCP	ACK	28083	ftp control	1224623295	2656441863	64572
2395	02:17:11:4105 GMT	78	137.72.43.207	137.72.43.110	TCP	ACK PSH : ftp reply code 250	ftp control	28127	2658535080	1101134151	32732
2397	02:17:11:5647 GMT	40	137.72.43.110	137.72.43.207	TCP	ACK	28127	ftp control	1101134151	2658535118	65194
2398	02:17:12:2098 GMT	46	137.72.43.110	137.72.43.207	TCP	ACK PSH : ftp command QUIT	28127	ftp control	1101134151	2658535118	65194
2399	02:17:12:2123 GMT	77	137.72.43.207	137.72.43.110	TCP	ACK PSH : ftp reply code 221	ftp control	28127	2658535118	1101134157	32762
2400	02:17:12:2148 GMT	40	137.72.43.207	137.72.43.110	TCP	ACK PSH FIN	ftp control	28127	2658535155	1101134157	32762
2401	02:17:12:2170 GMT	40	137.72.43.110	137.72.43.207	TCP	ACK FIN	28127	ftp control	1101134157	2658535155	65157
2402	02:17:12:2170 GMT	40	137.72.43.110	137.72.43.207	TCP	ACK	28127	ftp control	1101134158	2658535156	65157

Packet Details – PORT command



*Data Connection Port = 256*109 + 225 = 28129*

Control Connection (Passive FTP)

CleverView® for cTrace Analysis 5.1

File Help

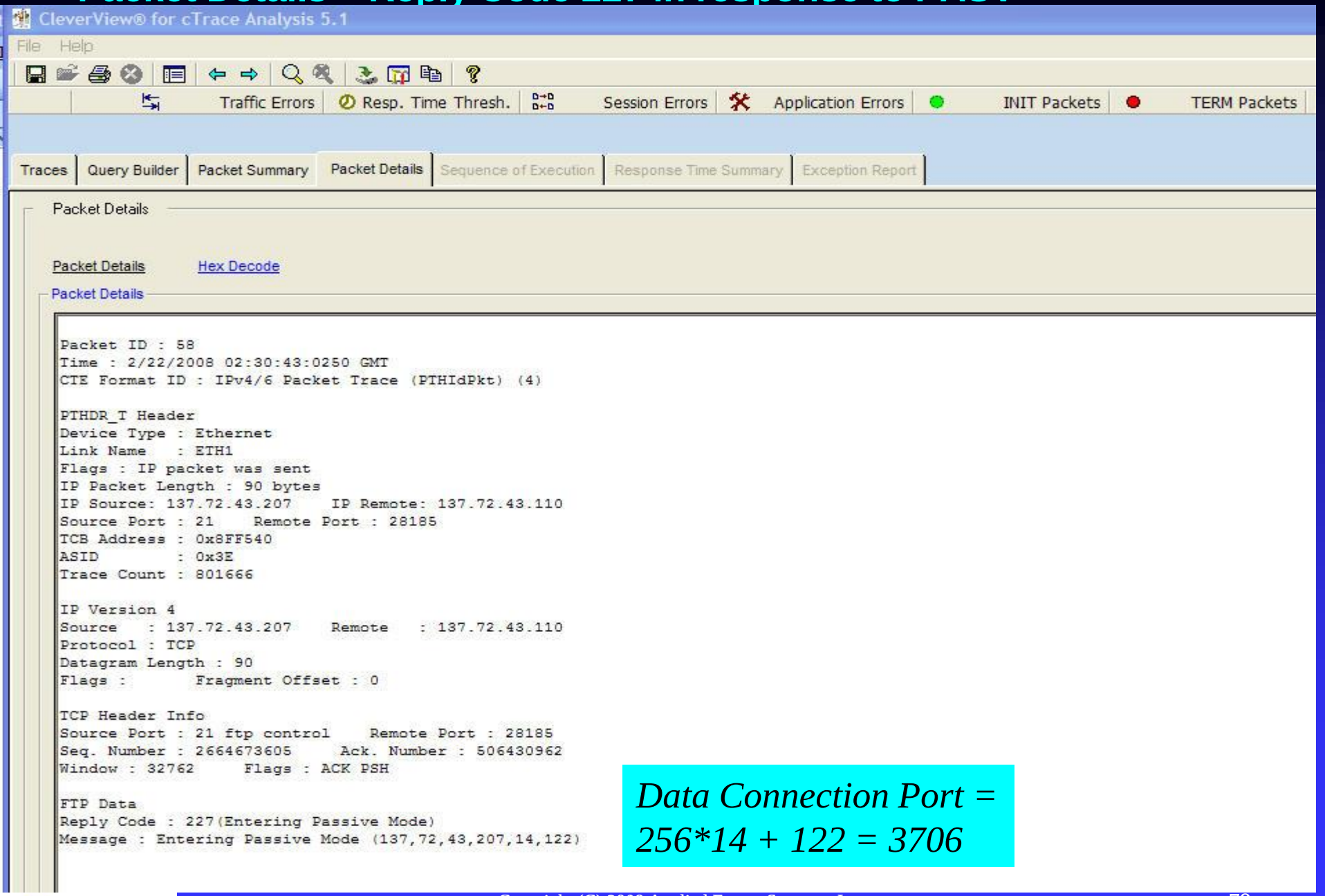
Traffic Errors | Resp. Time Thresh. | Session Errors | Application Errors | INIT Packets | TERM Packets

Traces | Query Builder | Packet Summary | Packet Details | Sequence of Execution | Response Time Summary | Exception Report

Packet Summary

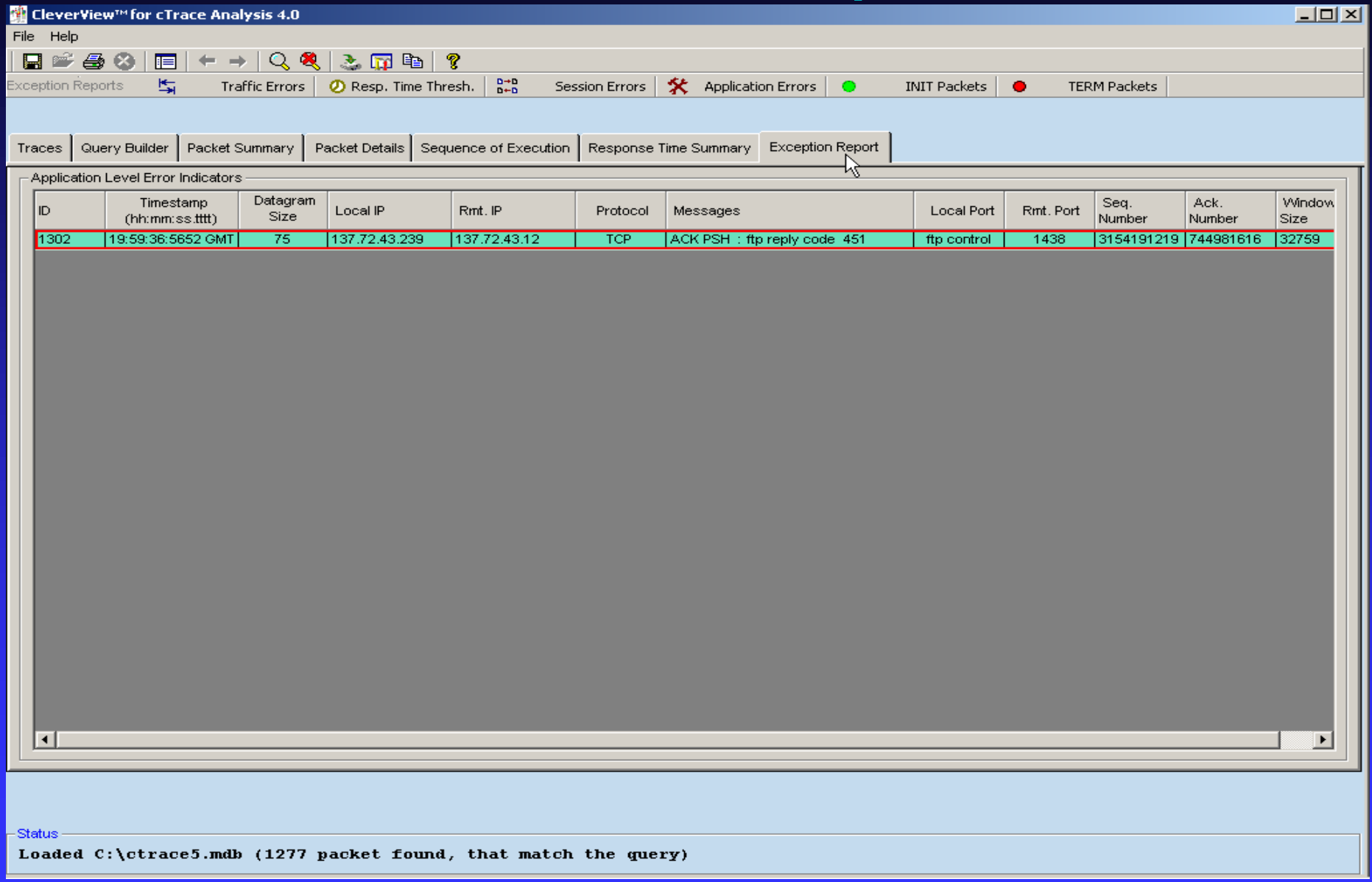
ID	Timestamp	Datagram Size	Local IP	Rmt. IP	Protocol	Messages	Local Port	Rmt. Port	Seq. Number	Ack. Number	Window Size
8	02:30:32:3970 GMT	40	137.72.43.110	137.72.43.207	TCP	ACK FIN	28180	ftp control	1125823318	2663902449	65032
9	02:30:32:4015 GMT	40	137.72.43.207	137.72.43.110	TCP	ACK PSH FIN	ftp control	28180	2663902449	1125823319	32760
10	02:30:32:4052 GMT	40	137.72.43.110	137.72.43.207	TCP	ACK	28180	ftp control	1125823319	2663902450	65032
11	02:30:32:5303 GMT	52	137.72.43.110	137.72.43.207	TCP	SYN	28185	ftp control	506430877	0	65534
12	02:30:32:5312 GMT	44	137.72.43.207	137.72.43.110	TCP	ACK SYN	ftp control	28185	2664673065	506430878	32768
13	02:30:32:5324 GMT	40	137.72.43.110	137.72.43.207	TCP	ACK	28185	ftp control	506430878	2664673066	65534
14	02:30:32:6692 GMT	114	137.72.43.207	137.72.43.110	TCP	ACK PSH : ftp reply code 220	ftp control	28185	2664673066	506430878	32768
15	02:30:32:8311 GMT	40	137.72.43.110	137.72.43.207	TCP	ACK	28185	ftp control	506430878	2664673140	65460
16	02:30:32:8316 GMT	74	137.72.43.207	137.72.43.110	TCP	ACK PSH : ftp reply code 220	ftp control	28185	2664673140	506430878	32768
17	02:30:32:8395 GMT	54	137.72.43.110	137.72.43.207	TCP	ACK PSH : ftp command USER	28185	ftp control	506430878	2664673174	65426
18	02:30:33:0706 GMT	67	137.72.43.207	137.72.43.110	TCP	ACK PSH : ftp reply code 331	ftp control	28185	2664673174	506430892	32754
19	02:30:33:0814 GMT	55	137.72.43.110	137.72.43.207	TCP	ACK PSH : ftp command PASS	28185	ftp control	506430892	2664673201	65399
20	02:30:33:3304 GMT	40	137.72.43.207	137.72.43.110	TCP	ACK PSH	ftp control	28185	2664673201	506430907	32753
21	02:30:33:4839 GMT	101	137.72.43.207	137.72.43.110	TCP	ACK PSH : ftp reply code 230	ftp control	28185	2664673201	506430907	32753
22	02:30:33:4911 GMT	46	137.72.43.110	137.72.43.207	TCP	ACK PSH	28185	ftp control	506430907	2664673262	65338
23	02:30:33:4943 GMT	69	137.72.43.207	137.72.43.110	TCP	ACK PSH : ftp reply code 211	ftp control	28185	2664673262	506430913	32762
24	02:30:33:4970 GMT	46	137.72.43.110	137.72.43.207	TCP	ACK PSH : ftp command SYST	28185	ftp control	506430913	2664673291	65309
25	02:30:33:5006 GMT	120	137.72.43.207	137.72.43.110	TCP	ACK PSH : ftp reply code 215	ftp control	28185	2664673291	506430919	32762
26	02:30:33:5567 GMT	45	137.72.43.110	137.72.43.207	TCP	ACK PSH : ftp command PWD	28185	ftp control	506430919	2664673371	65229
27	02:30:33:5590 GMT	80	137.72.43.207	137.72.43.110	TCP	ACK PSH : ftp reply code 257	ftp control	28185	2664673371	506430924	32763
28	02:30:33:7331 GMT	40	137.72.43.110	137.72.43.207	TCP	ACK	28185	ftp control	506430924	2664673411	65189
51	02:30:42:9803 GMT	59	137.72.43.110	137.72.43.207	TCP	ACK PSH : ftp command CWD	28185	ftp control	506430924	2664673411	65189
52	02:30:42:9942 GMT	117	137.72.43.207	137.72.43.110	TCP	ACK PSH : ftp reply code 250	ftp control	28185	2664673411	506430943	32749
53	02:30:42:9997 GMT	45	137.72.43.110	137.72.43.207	TCP	ACK PSH : ftp command PWD	28185	ftp control	506430943	2664673488	65112
54	02:30:43:0047 GMT	114	137.72.43.207	137.72.43.110	TCP	ACK PSH : ftp reply code 257	ftp control	28185	2664673488	506430948	32763
55	02:30:43:0102 GMT	48	137.72.43.110	137.72.43.207	TCP	ACK PSH : ftp command TYPE	28185	ftp control	506430948	2664673562	65038
56	02:30:43:0142 GMT	83	137.72.43.207	137.72.43.110	TCP	ACK PSH : ftp reply code 200	ftp control	28185	2664673562	506430956	32760
57	02:30:43:0208 GMT	46	137.72.43.110	137.72.43.207	TCP	ACK PSH : ftp command PASV	28185	ftp control	506430956	2664673605	64995
58	02:30:43:0250 GMT	90	137.72.43.207	137.72.43.110	TCP	ACK PSH : ftp reply code 227	ftp control	28185	2664673605	506430962	32762
59	02:30:43:0307 GMT	46	137.72.43.110	137.72.43.207	TCP	ACK PSH : ftp command LIST	28185	ftp control	506430962	2664673655	64945
63	02:30:43:1110 GMT	61	137.72.43.207	137.72.43.110	TCP	ACK PSH : ftp reply code 125	ftp control	28185	2664673655	506430968	32762
69	02:30:43:2633 GMT	40	137.72.43.110	137.72.43.207	TCP	ACK	28185	ftp control	506430968	2664673676	64924
70	02:30:43:2638 GMT	74	137.72.43.207	137.72.43.110	TCP	ACK PSH : ftp reply code 250	ftp control	28185	2664673676	506430968	32762
71	02:30:43:4639 GMT	40	137.72.43.110	137.72.43.207	TCP	ACK	28185	ftp control	506430968	2664673710	64890

Packet Details – Reply Code 227 in response to PASV

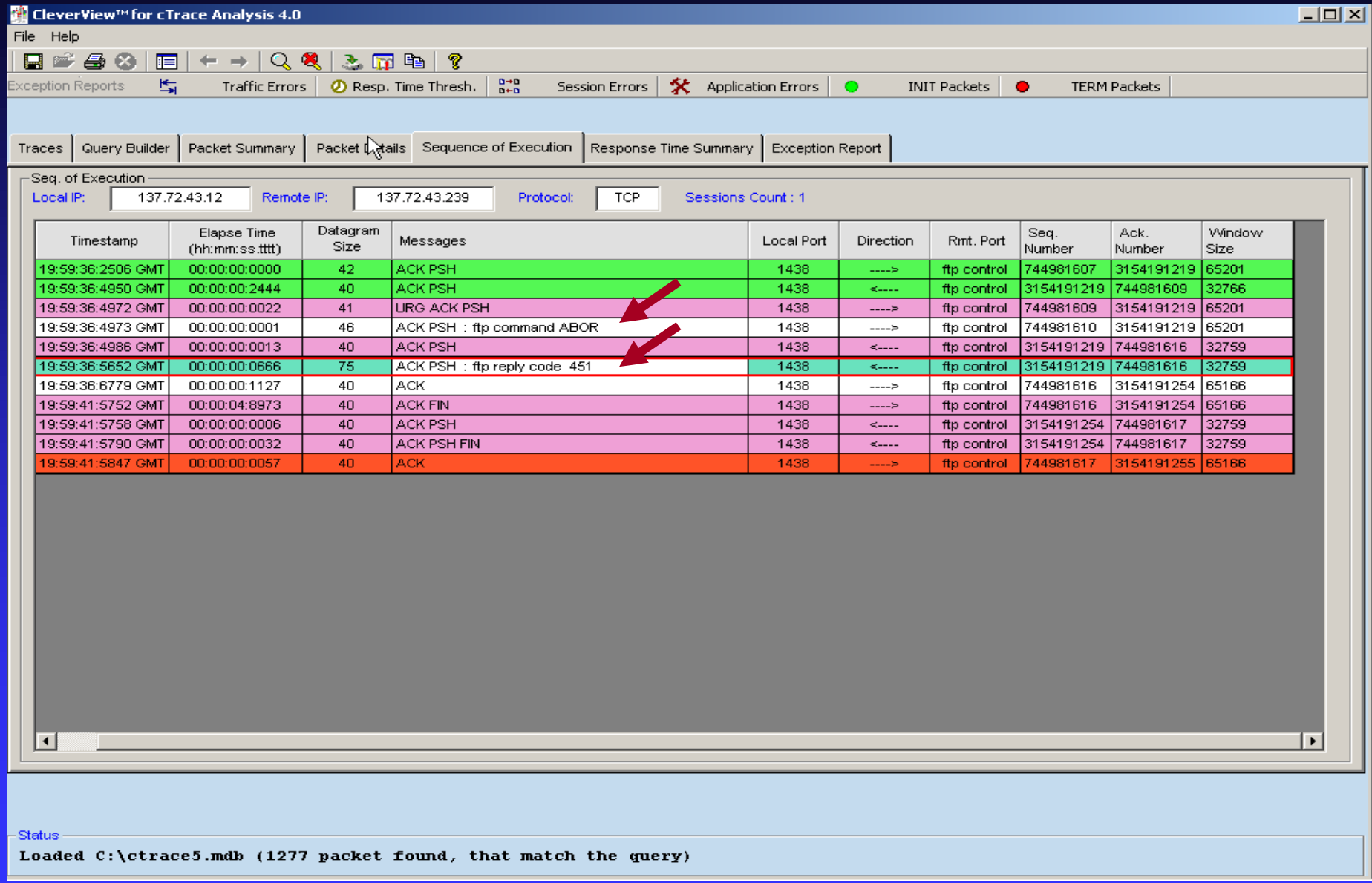


*Data Connection Port = 256*14 + 122 = 3706*

FTP Control Connection: exceptions

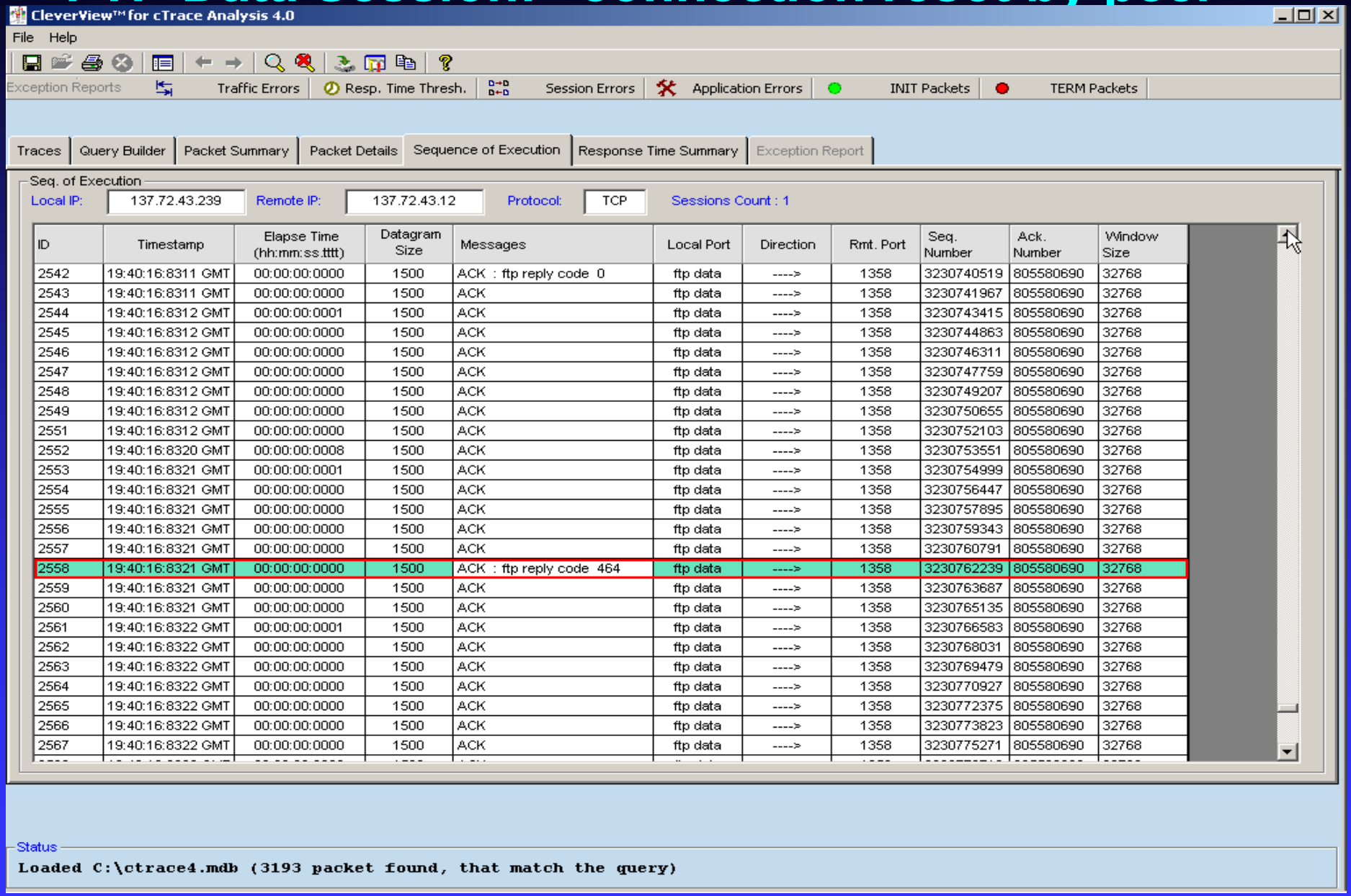


FTP Control Session: “451 transfer aborted”



Timestamp	Elapse Time (hh:mm:ss.tttt)	Datagram Size	Messages	Local Port	Direction	Rmt. Port	Seq. Number	Ack. Number	Window Size
19:59:36:2506 GMT	00:00:00:0000	42	ACK PSH	1438	---->	ftp control	744981607	3154191219	65201
19:59:36:4950 GMT	00:00:00:2444	40	ACK PSH	1438	<----	ftp control	3154191219	744981609	32766
19:59:36:4972 GMT	00:00:00:0022	41	URG ACK PSH	1438	---->	ftp control	744981609	3154191219	65201
19:59:36:4973 GMT	00:00:00:0001	46	ACK PSH : ftp command ABOR	1438	---->	ftp control	744981610	3154191219	65201
19:59:36:4986 GMT	00:00:00:0013	40	ACK PSH	1438	<----	ftp control	3154191219	744981616	32759
19:59:36:5652 GMT	00:00:00:0666	75	ACK PSH : ftp reply code 451	1438	<----	ftp control	3154191219	744981616	32759
19:59:36:6779 GMT	00:00:00:1127	40	ACK	1438	---->	ftp control	744981616	3154191254	65166
19:59:41:5752 GMT	00:00:04:8973	40	ACK FIN	1438	---->	ftp control	744981616	3154191254	65166
19:59:41:5758 GMT	00:00:00:0006	40	ACK PSH	1438	<----	ftp control	3154191254	744981617	32759
19:59:41:5790 GMT	00:00:00:0032	40	ACK PSH FIN	1438	<----	ftp control	3154191254	744981617	32759
19:59:41:5847 GMT	00:00:00:0057	40	ACK	1438	---->	ftp control	744981617	3154191255	65166

FTP Data Session: “connection reset by peer”



ID	Timestamp	Elapse Time (hh:mm:ss.tttt)	Datagram Size	Messages	Local Port	Direction	Rmt. Port	Seq. Number	Ack. Number	Window Size
2542	19:40:16:8311 GMT	00:00:00:0000	1500	ACK : ftp reply code 0	ftp data	---->	1358	3230740519	805580690	32768
2543	19:40:16:8311 GMT	00:00:00:0000	1500	ACK	ftp data	---->	1358	3230741967	805580690	32768
2544	19:40:16:8312 GMT	00:00:00:0001	1500	ACK	ftp data	---->	1358	3230743415	805580690	32768
2545	19:40:16:8312 GMT	00:00:00:0000	1500	ACK	ftp data	---->	1358	3230744863	805580690	32768
2546	19:40:16:8312 GMT	00:00:00:0000	1500	ACK	ftp data	---->	1358	3230746311	805580690	32768
2547	19:40:16:8312 GMT	00:00:00:0000	1500	ACK	ftp data	---->	1358	3230747759	805580690	32768
2548	19:40:16:8312 GMT	00:00:00:0000	1500	ACK	ftp data	---->	1358	3230749207	805580690	32768
2549	19:40:16:8312 GMT	00:00:00:0000	1500	ACK	ftp data	---->	1358	3230750655	805580690	32768
2551	19:40:16:8312 GMT	00:00:00:0000	1500	ACK	ftp data	---->	1358	3230752103	805580690	32768
2552	19:40:16:8320 GMT	00:00:00:0008	1500	ACK	ftp data	---->	1358	3230753551	805580690	32768
2553	19:40:16:8321 GMT	00:00:00:0001	1500	ACK	ftp data	---->	1358	3230754999	805580690	32768
2554	19:40:16:8321 GMT	00:00:00:0000	1500	ACK	ftp data	---->	1358	3230756447	805580690	32768
2555	19:40:16:8321 GMT	00:00:00:0000	1500	ACK	ftp data	---->	1358	3230757895	805580690	32768
2556	19:40:16:8321 GMT	00:00:00:0000	1500	ACK	ftp data	---->	1358	3230759343	805580690	32768
2557	19:40:16:8321 GMT	00:00:00:0000	1500	ACK	ftp data	---->	1358	3230760791	805580690	32768
2558	19:40:16:8321 GMT	00:00:00:0000	1500	ACK : ftp reply code 464	ftp data	---->	1358	3230762239	805580690	32768
2559	19:40:16:8321 GMT	00:00:00:0000	1500	ACK	ftp data	---->	1358	3230763687	805580690	32768
2560	19:40:16:8321 GMT	00:00:00:0000	1500	ACK	ftp data	---->	1358	3230765135	805580690	32768
2561	19:40:16:8322 GMT	00:00:00:0001	1500	ACK	ftp data	---->	1358	3230766583	805580690	32768
2562	19:40:16:8322 GMT	00:00:00:0000	1500	ACK	ftp data	---->	1358	3230768031	805580690	32768
2563	19:40:16:8322 GMT	00:00:00:0000	1500	ACK	ftp data	---->	1358	3230769479	805580690	32768
2564	19:40:16:8322 GMT	00:00:00:0000	1500	ACK	ftp data	---->	1358	3230770927	805580690	32768
2565	19:40:16:8322 GMT	00:00:00:0000	1500	ACK	ftp data	---->	1358	3230772375	805580690	32768
2566	19:40:16:8322 GMT	00:00:00:0000	1500	ACK	ftp data	---->	1358	3230773823	805580690	32768
2567	19:40:16:8322 GMT	00:00:00:0000	1500	ACK	ftp data	---->	1358	3230775271	805580690	32768

FTP Data Session: “connection reset by peer”

CleverView™ for cTrace Analysis 4.0

File Help

Exception Reports | Traffic Errors | Resp. Time Thresh. | Session Errors | Application Errors | INIT Packets | TERM Packets

Traces | Query Builder | Packet Summary | Packet Details | Sequence of Execution | Response Time Summary | Exception Report

Seq. of Execution

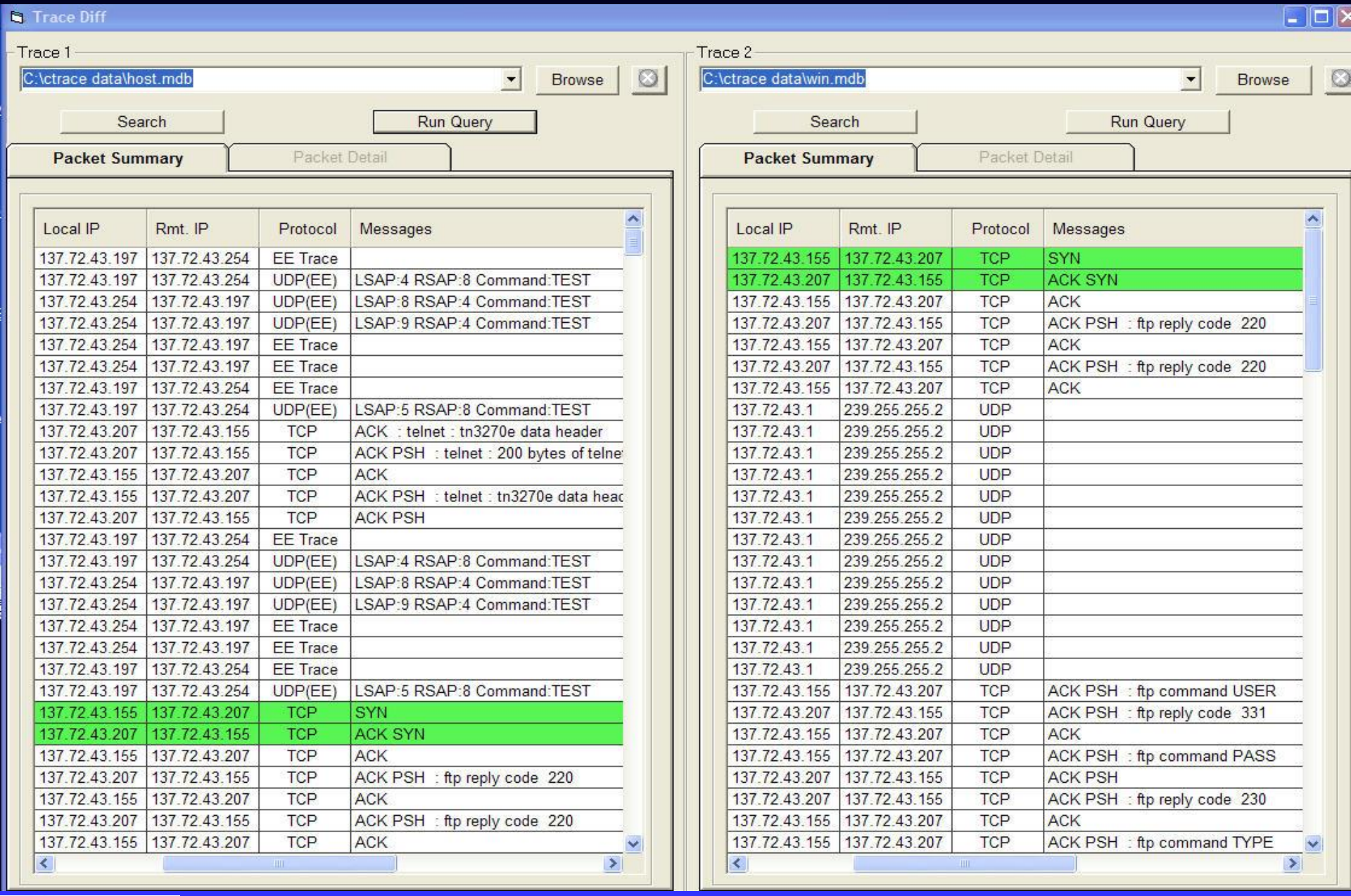
Local IP: 137.72.43.239 Remote IP: 137.72.43.12 Protocol: TCP Sessions Count : 1

ID	Timestamp	Elapse Time (hh:mm:ss.tttt)	Datagram Size	Messages	Local Port	Direction	Rmt. Port	Seq. Number	Ack. Number	Window Size
2710	19:40:16:8603 GMT	00:00:00:0000	1500	ACK	ftp data	---->	1358	3230904073	805580690	32768
2711	19:40:16:8603 GMT	00:00:00:0000	1500	ACK	ftp data	---->	1358	3230905521	805580690	32768
2712	19:40:16:8604 GMT	00:00:00:0001	1500	ACK	ftp data	---->	1358	3230906969	805580690	32768
2713	19:40:16:8604 GMT	00:00:00:0000	1500	ACK	ftp data	---->	1358	3230908417	805580690	32768
2714	19:40:16:8604 GMT	00:00:00:0000	1500	ACK	ftp data	---->	1358	3230909865	805580690	32768
2715	19:40:16:8604 GMT	00:00:00:0000	1500	ACK	ftp data	---->	1358	3230911313	805580690	32768
2716	19:40:16:8604 GMT	00:00:00:0000	1500	ACK	ftp data	---->	1358	3230912761	805580690	32768
2717	19:40:16:8604 GMT	00:00:00:0000	1500	ACK	ftp data	---->	1358	3230914209	805580690	32768
2718	19:40:16:8604 GMT	00:00:00:0000	1500	ACK	ftp data	---->	1358	3230915657	805580690	32768
2719	19:40:16:8604 GMT	00:00:00:0000	1500	ACK	ftp data	---->	1358	3230917105	805580690	32768
2720	19:40:16:8605 GMT	00:00:00:0001	1500	ACK	ftp data	---->	1358	3230918553	805580690	32768
2721	19:40:16:8605 GMT	00:00:00:0000	1500	ACK	ftp data	---->	1358	3230920001	805580690	32768
2723	19:40:16:8605 GMT	00:00:00:0000	872	ACK PSH	ftp data	---->	1358	3230921449	805580690	32768
2735	19:40:17:4727 GMT	00:00:00:6122	1500	ACK PSH	ftp data	---->	1358	3230856734	805580690	32768
2737	19:40:18:1233 GMT	00:00:00:6506	1500	ACK PSH	ftp data	---->	1358	3230856734	805580690	32768
2757	19:40:18:7834 GMT	00:00:00:6601	1500	ACK PSH	ftp data	---->	1358	3230856734	805580690	32768
2758	19:40:19:4431 GMT	00:00:00:6597	1500	ACK PSH	ftp data	---->	1358	3230856734	805580690	32768
2759	19:40:20:0926 GMT	00:00:00:6495	1500	ACK PSH	ftp data	---->	1358	3230856734	805580690	32768
2760	19:40:20:7436 GMT	00:00:00:6510	1500	ACK PSH	ftp data	---->	1358	3230856734	805580690	32768
2762	19:40:21:9435 GMT	00:00:01:1999	1500	ACK PSH	ftp data	---->	1358	3230856734	805580690	32768
2767	19:40:24:2132 GMT	00:00:02:2697	1500	ACK PSH	ftp data	---->	1358	3230856734	805580690	32768
2776	19:40:28:6929 GMT	00:00:04:4797	1500	ACK PSH	ftp data	---->	1358	3230856734	805580690	32768
2791	19:40:37:6132 GMT	00:00:08:9203	1500	ACK PSH	ftp data	---->	1358	3230856734	805580690	32768
2822	19:40:55:4130 GMT	00:00:17:7998	1500	ACK PSH	ftp data	---->	1358	3230856734	805580690	32768
2957	19:41:31:0041 GMT	00:00:35:5911	1500	ACK PSH	ftp data	---->	1358	3230856734	805580690	32768

Status

Loaded C:\ctrace4.mdb (3193 packet found, that match the query)

Trace Comparison – Host vs. Windows



Local IP	Rmt. IP	Protocol	Messages
137.72.43.197	137.72.43.254	EE Trace	
137.72.43.197	137.72.43.254	UDP(EE)	LSAP:4 RSAP:8 Command:TEST
137.72.43.254	137.72.43.197	UDP(EE)	LSAP:8 RSAP:4 Command:TEST
137.72.43.254	137.72.43.197	UDP(EE)	LSAP:9 RSAP:4 Command:TEST
137.72.43.254	137.72.43.197	EE Trace	
137.72.43.254	137.72.43.197	EE Trace	
137.72.43.197	137.72.43.254	EE Trace	
137.72.43.197	137.72.43.254	UDP(EE)	LSAP:5 RSAP:8 Command:TEST
137.72.43.207	137.72.43.155	TCP	ACK : telnet : tn3270e data header
137.72.43.207	137.72.43.155	TCP	ACK PSH : telnet : 200 bytes of telne
137.72.43.155	137.72.43.207	TCP	ACK
137.72.43.155	137.72.43.207	TCP	ACK PSH : telnet : tn3270e data heac
137.72.43.207	137.72.43.155	TCP	ACK PSH
137.72.43.197	137.72.43.254	EE Trace	
137.72.43.197	137.72.43.254	UDP(EE)	LSAP:4 RSAP:8 Command:TEST
137.72.43.254	137.72.43.197	UDP(EE)	LSAP:8 RSAP:4 Command:TEST
137.72.43.254	137.72.43.197	UDP(EE)	LSAP:9 RSAP:4 Command:TEST
137.72.43.254	137.72.43.197	EE Trace	
137.72.43.254	137.72.43.197	EE Trace	
137.72.43.197	137.72.43.254	EE Trace	
137.72.43.197	137.72.43.254	UDP(EE)	LSAP:5 RSAP:8 Command:TEST
137.72.43.155	137.72.43.207	TCP	SYN
137.72.43.207	137.72.43.155	TCP	ACK SYN
137.72.43.155	137.72.43.207	TCP	ACK
137.72.43.207	137.72.43.155	TCP	ACK PSH : ftp reply code 220
137.72.43.155	137.72.43.207	TCP	ACK
137.72.43.207	137.72.43.155	TCP	ACK PSH : ftp reply code 220
137.72.43.155	137.72.43.207	TCP	ACK

Local IP	Rmt. IP	Protocol	Messages
137.72.43.155	137.72.43.207	TCP	SYN
137.72.43.207	137.72.43.155	TCP	ACK SYN
137.72.43.155	137.72.43.207	TCP	ACK
137.72.43.207	137.72.43.155	TCP	ACK PSH : ftp reply code 220
137.72.43.155	137.72.43.207	TCP	ACK
137.72.43.207	137.72.43.155	TCP	ACK PSH : ftp reply code 220
137.72.43.155	137.72.43.207	TCP	ACK
137.72.43.1	239.255.255.2	UDP	
137.72.43.1	239.255.255.2	UDP	
137.72.43.1	239.255.255.2	UDP	
137.72.43.1	239.255.255.2	UDP	
137.72.43.1	239.255.255.2	UDP	
137.72.43.1	239.255.255.2	UDP	
137.72.43.1	239.255.255.2	UDP	
137.72.43.1	239.255.255.2	UDP	
137.72.43.1	239.255.255.2	UDP	
137.72.43.1	239.255.255.2	UDP	
137.72.43.1	239.255.255.2	UDP	
137.72.43.1	239.255.255.2	UDP	
137.72.43.1	239.255.255.2	UDP	
137.72.43.1	239.255.255.2	UDP	
137.72.43.155	137.72.43.207	TCP	ACK PSH : ftp command USER
137.72.43.207	137.72.43.155	TCP	ACK PSH : ftp reply code 331
137.72.43.155	137.72.43.207	TCP	ACK
137.72.43.155	137.72.43.207	TCP	ACK PSH : ftp command PASS
137.72.43.207	137.72.43.155	TCP	ACK PSH
137.72.43.207	137.72.43.155	TCP	ACK PSH : ftp reply code 230
137.72.43.155	137.72.43.207	TCP	ACK
137.72.43.155	137.72.43.207	TCP	ACK PSH : ftp command TYPE

FTP Diagnosis of Data Connection Issues

- Obtain a packet or OSAENTA trace.
- Obtain a server trace with options BAS and FLO.
  - F ftpx,DEBUG=(FLO,BAS)
- Find the PORT or PASV command.
- Determine the IP address and port number for data connection.
- Check for session establishment: SYN, SYN ACK, etc.
- Missing SYN ACK may indicate firewall issue; e.g., for Passive FTP:
  - FTP.DATA - PASSIVEDATAPORTS – should match firewall range
  - TCP PROFILE – PORTRANGE
  - E.g., firewall allows 5000-6000;
    - PASSIVEPORTS(5000,6000)
    - PORTRANGE 5000 1001 AUTHPORT

FTP Tuning

- Dispatching priorities of the FTP Server and the user task
  - FTP Server uses fork() to create a new address for each connection – ensure each forked address space will receive enough CPU
  - The FTP client will run with the same dispatching priority as the TSO userid who is logged on
- To determine if the bottleneck is DASD related, test network throughput only without disk I/O:
  - Prior to PUT: **cd *dev.null**
  - Prior to GET: **lcd *dev.null**

FTP Tuning

- Use the right Data Type (EBCDIC vs. ASCII)
- TCP Window Size: the maximum amount of data that can be in the network at any time for a single connection.
- Optimal TCP Window Size =

  Bottleneck Bandwidth * Round-trip Time (RTT)
- E.g., the *slowest* link=45 Mbit/sec, RTT=20ms

  45 Mbit/sec * 20ms

  = 45,000,000 bits/sec * .020 sec

  = 900,000 bits = 109.86 KBytes

FTP Tuning

- RTT
  - Ping with default packet size; e.g., 256
  - Ping with “average” FTP packet size
  - SMF 119 TCP Connection Termination Record (RTT *at time* of connection close)
  - Packet trace
- Window Size
  - SMF 119 TCP Connection Termination Record
  - Packet trace
  - D NETSTAT,CONFIG
  - TCPCONFIG TCPSENDBFRSIZE 64K TCPRCVBUFRSIZE 64K
  - Windows tools: Dr. TCP, TCP Optimizer, etc.

FTP Tuning

- FTP.DATA
  - Ping with default packet size; e.g., 256
  - Ping with “average” FTP packet size
  - SMF 119 TCP Connection Termination Record (RTT *at time* of connection close)
  - Packet trace
- PROFILE.TCPIP
  - NODELAYACKS (on FTP ports, or on the route); by default, TCP waits 200ms to send an ACK
  - TCPCONFIG
    - TCPSENDBFRSIZE : 64K or greater (default=16K)
    - TCPRCVBUFRSIZE : 64K or greater (default=16K)
    - TCPMAXRCVBUFRSIZE : minimum 180K, default=256K

FTP Analysis

- Global usage patterns:
  - Total FTP sessions
  - Total FTP bytes
  - FTP server vs. FTP client activities
  - When are heavy FTPs done?
  - Are gigabyte file transfers done?
  - Unauthorized attempts – logon failure reason
- Heavy users: Typically may be responsible for 80% of the workload
  - Which data sets are most heavily used?
  - Certain type? SEQ/ JES/ SQL?
  - Can reposition or copy data sets for better performance?
- Heavy data set usage: Often moved around or duplicates made for security and other redundancy reasons
- FTP performance analysis: throughput and response time

FTP Analysis

- FTP Failure Analysis – Who and Why
  - When do failures occur? At a specific time of day?
  - Who are the top failing clients and datasets?
  - Failure with one dataset or all datasets?
  - Logon failure vs. data transfer failure
  - Correlated with heavy usage?
  - Client side or server side – or in the middle?
  - Control connection or data connection?
  - Direction of flow – try reversing the role of client/server

- FTP Historical Trending and Analysis
  - Number of server/client sessions
  - Types of transfers (SEQ/SQL/JES)
  - Amount of transfers
  - Failures
  - Heavy usage hours
  - Throughput
  - Workload analysis = proactive problem diagnostics