FTP Transfer (FTP Session Sequence Diagram)

Internet			FTP Client PC					EventHelix.com/EventStudio 2.5
DNS	FTP Server		FTP Client				Console	
Port 53	Port 20	Port 21	1030	1175	5001	5002	User	18-Mar-05 06:45 (Page 1)

This sequence diagram was generated with EventStudio 2.5 (http://www.EventHelix.com/EventStudio).

Here we explore the sequence of interactions in a typical FTP (File Transfer Protocol) session. The example here illustrates the use of multiple TCP connections by FTP. We will cover how FTP establishes a telnet TCP connection (TCP Port 21) to control the overall flow of the FTP transfer. Then we examine the use of TCP Port 21 for establishing TCP connections for directory transfer and file retrieval.

The complete sequence diagram can be divided into the following steps:
- DNS Query to obtain the IP address for the FTP Server
- FTP Telnet connection setup and login. (USER and PASS commands)
- Obtaining a directory listing (PORT and LIST command)
- Changing directory (CWD command)
- Downloading a file using FTP get (PORT and RETR command)

Copyright © 2005 EventHelix.com Inc. All Rights Reserved.

DNS Query to obtain the IP address for the FTP Server.

Invoke FTP client with ftp.any-domain.com

DNS Query
recdesired = 1, queries = 1, name = ftp.any-domain.com, type = A, class = IN

The FTP client sends a DNS query for domain name ftp.any-domain.com

DNS Response
recdesired = 1, answers = 1, IP Address

The DNS responds back with the IP address.

FTP Telnet connection setup and login. (USER and PASS commands).

TCP SYN
srcport = 1175, dstport = 21, syn = 1

FTP client initiates establishment of the telnet session TCP connection by sending a SYN to TCP port 21

TCP SYN+ACK
srcport = 21, dstport = 1175, syn = 1, ack = 1

The FTP server responds with SYN+ACK

TCP ACK
srcport = 1175, dstport = 21, ack = 1

The client machine responds with an ACK, this completes the TCP three way handshake.

FTP 220
ftp.response.code = 220, ftp.response.arg = ANY-DOMAIN.COM FTP Service

The FTP server then sends a 220 response to indicate that the FTP server is ready to accept a login.

Display welcome message

TCP ACK
srcport = 1175, dstport = 21, ack = 1

The client machine acknowledges the FTP 220 TCP packet.

Enter login information

USER ftp

User's login name is transported in a TCP segment. In this example an anonymous FTP is being initiated with the user name "ftp"

TCP ACK
srcport = 21, dstport = 1175, ack = 1

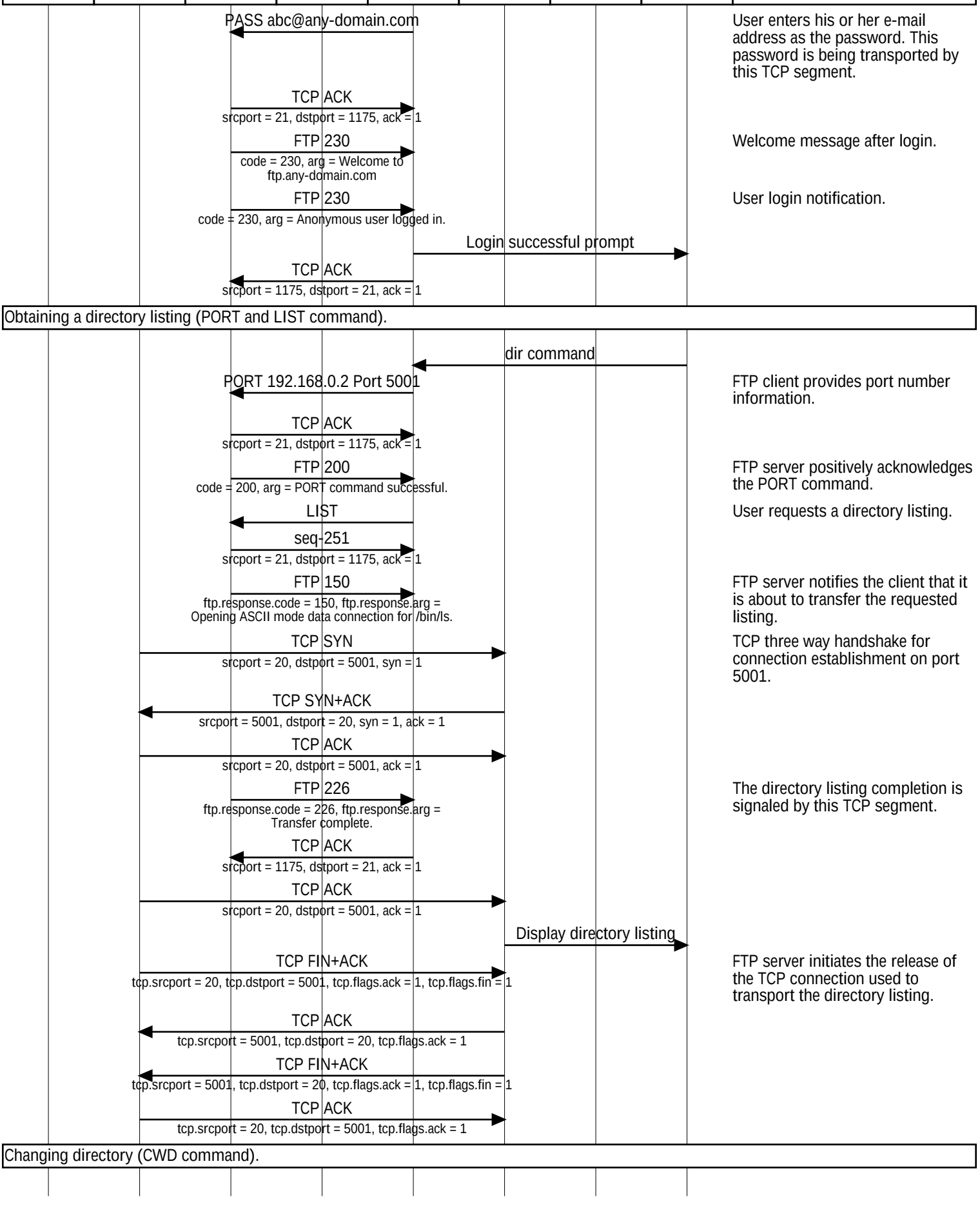
FTP 331
code = 331, arg = Anonymous access allowed, send identity (e-mail name) as password.

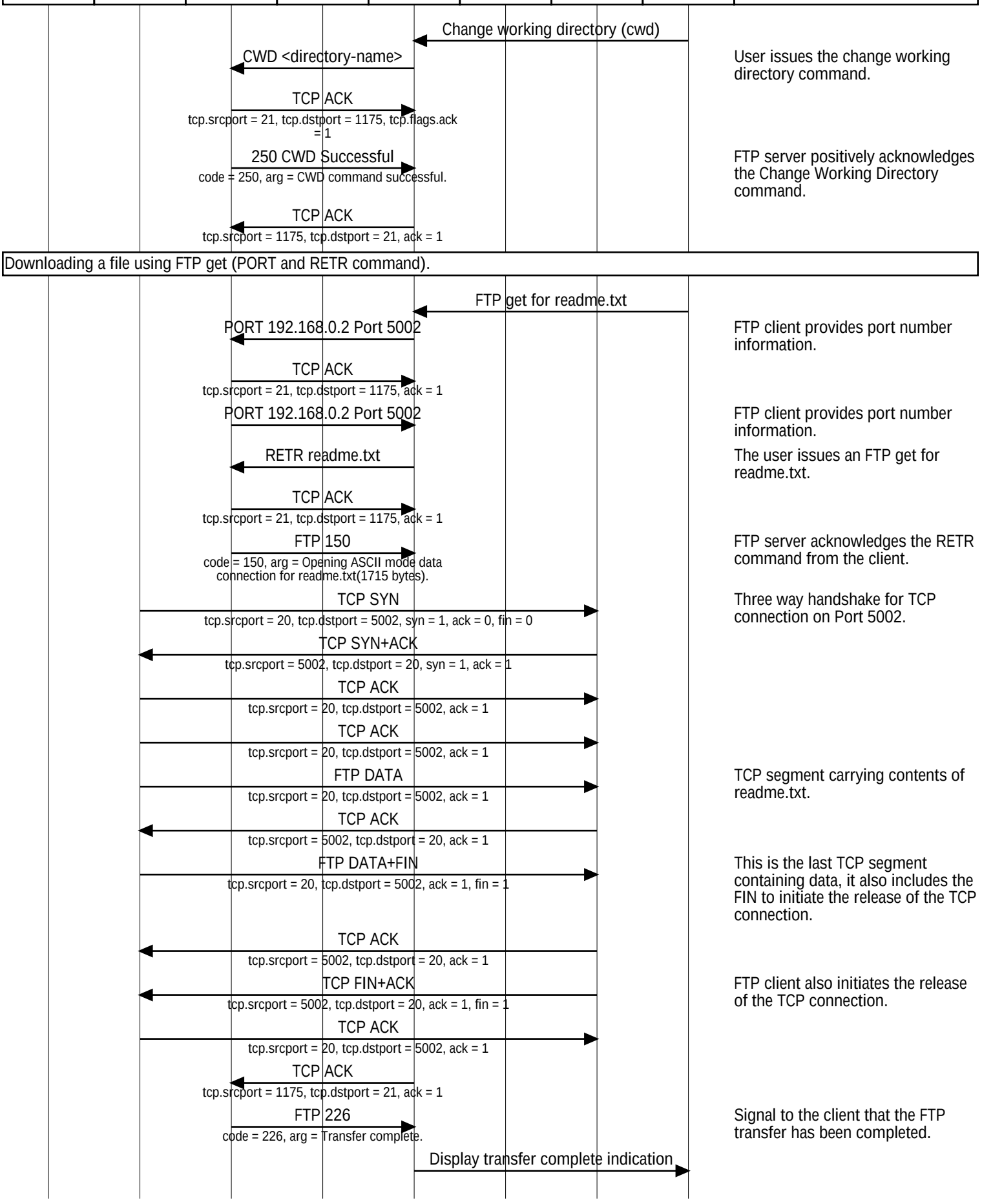
FTP server indicates that anonymous FTP is allowed.

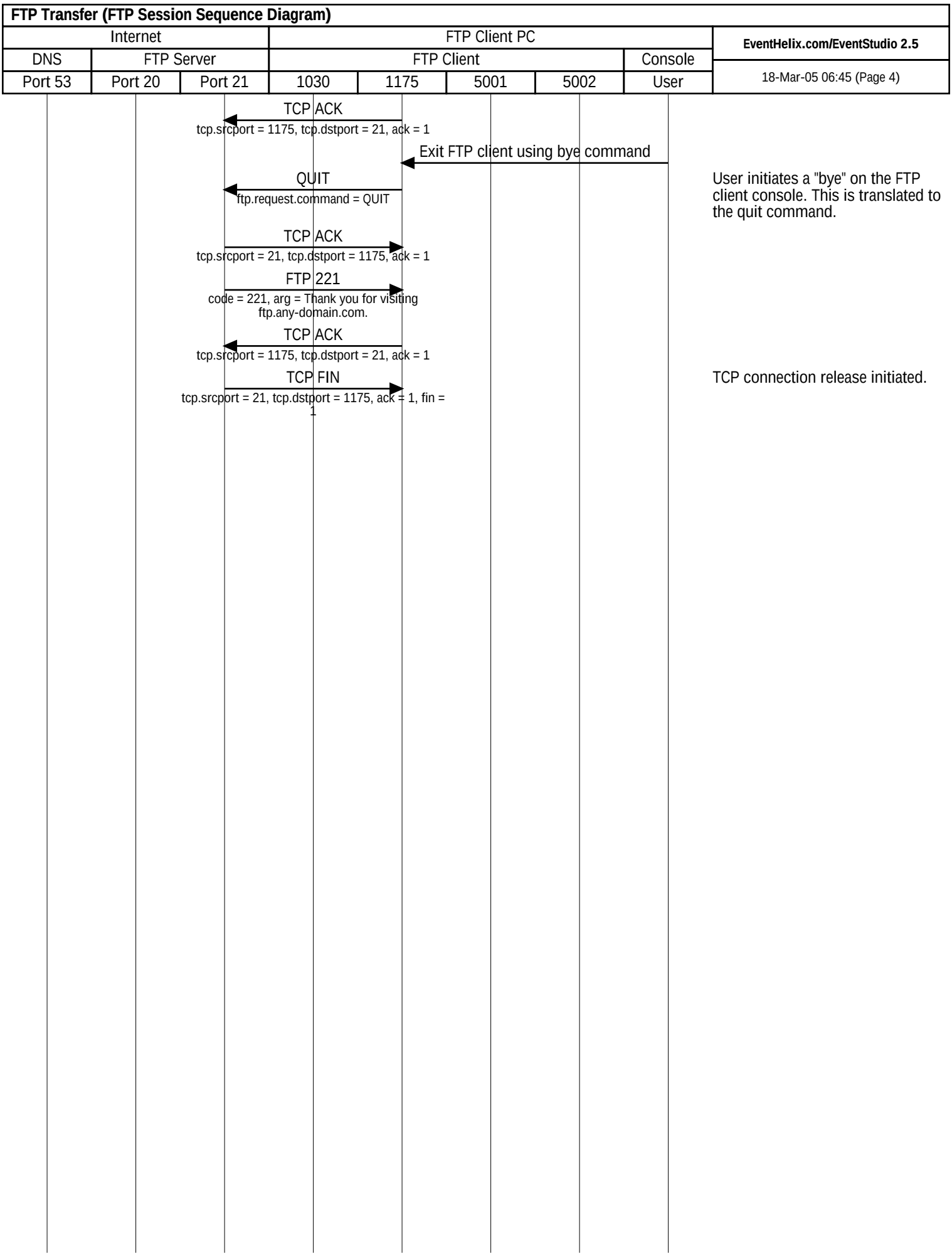
Password prompt

TCP ACK
srcport = 1175, dstport = 21, ack = 1

Password information


PASS abc@any-domain.com
User enters his or her e-mail address as the password. This password is being transported by this TCP segment.
TCP ACK
srcport = 21, dstport = 1175, ack = 1
FTP 230
code = 230, arg = Welcome to ftp.any-domain.com
Welcome message after login.
FTP 230
code = 230, arg = Anonymous user logged in.
User login notification.
Login successful prompt
TCP ACK
srcport = 1175, dstport = 21, ack = 1
Obtaining a directory listing (PORT and LIST command).
dir command
PORT 192.168.0.2 Port 5001
FTP client provides port number information.
TCP ACK
srcport = 21, dstport = 1175, ack = 1
FTP 200
code = 200, arg = PORT command successful.
FTP server positively acknowledges the PORT command.
LIST
User requests a directory listing.
seq-251
srcport = 21, dstport = 1175, ack = 1
FTP 150
ftp.response.code = 150, ftp.response.arg = Opening ASCII mode data connection for /bin/ls.
FTP server notifies the client that it is about to transfer the requested listing.
TCP SYN
srcport = 20, dstport = 5001, syn = 1
TCP three way handshake for connection establishment on port 5001.
TCP SYN+ACK
srcport = 5001, dstport = 20, syn = 1, ack = 1
TCP ACK
srcport = 20, dstport = 5001, ack = 1
FTP 226
ftp.response.code = 226, ftp.response.arg = Transfer complete.
The directory listing completion is signaled by this TCP segment.
TCP ACK
srcport = 1175, dstport = 21, ack = 1
TCP ACK
srcport = 20, dstport = 5001, ack = 1
Display directory listing
TCP FIN+ACK
tcp.srcport = 20, tcp.dstport = 5001, tcp.flags.ack = 1, tcp.flags.fin = 1
FTP server initiates the release of the TCP connection used to transport the directory listing.
TCP ACK
tcp.srcport = 5001, tcp.dstport = 20, tcp.flags.ack = 1
TCP FIN+ACK
tcp.srcport = 5001, tcp.dstport = 20, tcp.flags.ack = 1, tcp.flags.fin = 1
TCP ACK
tcp.srcport = 20, tcp.dstport = 5001, tcp.flags.ack = 1
Changing directory (CWD command).


Change working directory (cwd)
CWD <directory-name>
User issues the change working directory command.
TCP ACK
tcp.srcport = 21, tcp.dstport = 1175, tcp.flags.ack = 1
250 CWD Successful
code = 250, arg = CWD command successful.
FTP server positively acknowledges the Change Working Directory command.
TCP ACK
tcp.srcport = 1175, tcp.dstport = 21, ack = 1
Downloading a file using FTP get (PORT and RETR command).
FTP get for readme.txt
PORT 192.168.0.2 Port 5002
FTP client provides port number information.
TCP ACK
tcp.srcport = 21, tcp.dstport = 1175, ack = 1
PORT 192.168.0.2 Port 5002
FTP client provides port number information.
RETR readme.txt
The user issues an FTP get for readme.txt.
TCP ACK
tcp.srcport = 21, tcp.dstport = 1175, ack = 1
FTP 150
code = 150, arg = Opening ASCII mode data connection for readme.txt(1715 bytes).
FTP server acknowledges the RETR command from the client.
TCP SYN
tcp.srcport = 20, tcp.dstport = 5002, syn = 1, ack = 0, fin = 0
Three way handshake for TCP connection on Port 5002.
TCP SYN+ACK
tcp.srcport = 5002, tcp.dstport = 20, syn = 1, ack = 1
TCP ACK
tcp.srcport = 20, tcp.dstport = 5002, ack = 1
TCP ACK
tcp.srcport = 20, tcp.dstport = 5002, ack = 1
FTP DATA
tcp.srcport = 20, tcp.dstport = 5002, ack = 1
TCP segment carrying contents of readme.txt.
TCP ACK
tcp.srcport = 5002, tcp.dstport = 20, ack = 1
FTP DATA+FIN
tcp.srcport = 20, tcp.dstport = 5002, ack = 1, fin = 1
This is the last TCP segment containing data, it also includes the FIN to initiate the release of the TCP connection.
TCP ACK
tcp.srcport = 5002, tcp.dstport = 20, ack = 1
TCP FIN+ACK
tcp.srcport = 5002, tcp.dstport = 20, ack = 1, fin = 1
FTP client also initiates the release of the TCP connection.
TCP ACK
tcp.srcport = 20, tcp.dstport = 5002, ack = 1
TCP ACK
tcp.srcport = 1175, tcp.dstport = 21, ack = 1
FTP 226
code = 226, arg = Transfer complete.
Signal to the client that the FTP transfer has been completed.
Display transfer complete indication


FTP Transfer (FTP Session Sequence Diagram)
Internet
FTP Client PC
EventHelix.com/EventStudio 2.5
DNS
FTP Server
FTP Client
Console
Port 53
Port 20
Port 21
1030
1175
5001
5002
User
18-Mar-05 06:45 (Page 4)
TCP ACK
tcp.srcport = 1175, tcp.dstport = 21, ack = 1
Exit FTP client using bye command
QUIT
ftp.request.command = QUIT
User initiates a "bye" on the FTP client console. This is translated to the quit command.
TCP ACK
tcp.srcport = 21, tcp.dstport = 1175, ack = 1
FTP 221
code = 221, arg = Thank you for visiting ftp.any-domain.com.
TCP ACK
tcp.srcport = 1175, tcp.dstport = 21, ack = 1
TCP FIN
tcp.srcport = 21, tcp.dstport = 1175, ack = 1, fin = 1
TCP connection release initiated.