Laura Chappell

Analyzing FTP Communications

One of the advantages of being a protocol analyst is the ability to look at packet-level communications to see how an application operates. When errors occur, comparing the faulty communication sequences to the proper communication sequences helps you quickly identify the problem.

Since NetWare 5.1 and other servers include an FTP server daemon, you may be running FTP applications on your company's network. To help you troubleshoot these applications and their communications, this article examines FTP operations, focusing on the connection sequence, port definition, remote file listing, and file transfer. This article also lists the FTP reply codes and the FTP filters you should build to analyze your company's FTP communications.

To provide FTP communications for this article, I performed a lab test using BisonWare's FTP server software on a Windows 98 workstation. (The FTP server software runs as a Windows application.) From the FTP client, I simply used the ftp.exe that is included with Windows 98.

OVERVIEW OF FTP COMMUNICATIONS

FTP relies on TCP to provide a connection-oriented method for transferring files. The basic FTP application is defined to provide a simple, effective, and reliable method for transferring files between hosts. In most companies, however, the process of transferring files is simply one element of a more sophisticated and robust resource-sharing operating system, such as NetWare 5.1.

FTP is detailed in Request for Comments (RFC) 959 written by Postel and Reynolds in October 1985. The initial drafting for FTP, however, was written in 1971 in RFC 114. (You can download these RFCs from www.ietf.org/rfc.)

BASIC FTP COMMUNICATIONS

When you start filtering on and capturing FTP traffic on your company's network, you will notice that FTP traffic can have the following characteristics:

- FTP uses TCP.
- FTP uses port 21 for FTP commands.
- FTP can use a number of ephemeral (temporary/dynamic) ports for data transfer.
- FTP can use cleartext usernames and passwords.

Figure 1 shows a summary of an FTP connection and log out sequence. (See p. 24.) The FTP client's IP address is 10.2.0.2. The FTP server's IP address is 10.2.0.1.

Get the Trace! To view the complete communication sequence, download the ftp-connect.zip file from the Traces section at www.packet-level.com. This file contains the ftp-connect.cap trace file and the ftp-connect.pdf file.

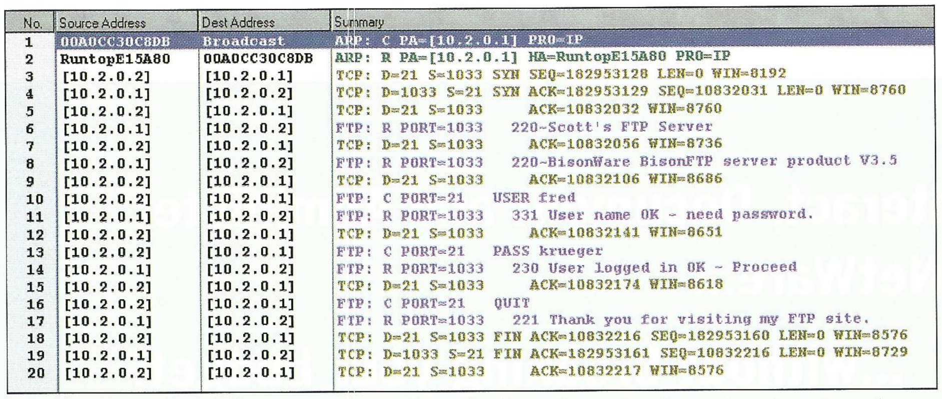
The process of establishing an FTP client connection with an FTP server and then logging out from that server includes five main steps. (See Figure 1 on p. 24.)

1. Use ARP to obtain hardware address (packets 1-2). The first two packets in the summary1.cap trace file show how the FTP client uses the ARP process to obtain the hardware address of the local FTP server.
2. Establish TCP connection (packets 3-5). Because FTP is defined as a connection-oriented protocol, the FTP client must perform a TCP handshake with the FTP server.

   The FTP client initiates the TCP connection, and then the FTP client and the FTP server exchange their starting TCP sequence number and their TCP window-size information. The TCP window dictates how much data can be outstanding on the wire before requiring an acknowledgment. During this handshake, the FTP client uses a dynamic port number for the source. (For more information about the TCP handshake, see "Inside the TCP Handshake," *NetWare Connection*, Mar. 2000, pp. 34–35. You can download this article from www.nwconnection.com/past.)
3. Log in and authenticate (packets 6-15). FTP servers typically send a login and logout message to FTP clients.

   FTP servers also require a username and password for each login. In packets 6 and 8, you can see two separate messages identifying the owner of the FTP server (Scott) and the FTP server product name and version number.

Illustration: Norman Felchle



```
No. Source Address  Dest Address   Summary
 1  00A0CC30C8DB    Broadcast      ARP: C PA=[10.2.0.1] PRO=IP
 2  RuntopE15A80    00A0CC30C8DB   ARP: R PA=[10.2.0.1] HA=RuntopE15A80 PRO=IP
 3  [10.2.0.2]      [10.2.0.1]     TCP: D=21 S=1033 SYN SEQ=182953128 LEN=0 WIN=8192
 4  [10.2.0.1]      [10.2.0.2]     TCP: D=1033 S=21 SYN ACK=182953129 SEQ=10832031 LEN=0 WIN=8760
 5  [10.2.0.2]      [10.2.0.1]     TCP: D=21 S=1033     ACK=10832032 WIN=8760
 6  [10.2.0.1]      [10.2.0.2]     FTP: R PORT=1033   220-Scott's FTP Server
 7  [10.2.0.2]      [10.2.0.1]     TCP: D=21 S=1033     ACK=10832056 WIN=8736
 8  [10.2.0.1]      [10.2.0.2]     FTP: R PORT=1033   220-BisonWare BisonFTP server product V3.5
 9  [10.2.0.2]      [10.2.0.1]     TCP: D=21 S=1033     ACK=10832106 WIN=8686
10  [10.2.0.2]      [10.2.0.1]     FTP: C PORT=21   USER fred
11  [10.2.0.1]      [10.2.0.2]     FTP: R PORT=1033   331 User name OK - need password.
12  [10.2.0.2]      [10.2.0.1]     TCP: D=21 S=1033     ACK=10832141 WIN=8651
13  [10.2.0.2]      [10.2.0.1]     FTP: C PORT=21   PASS krueger
14  [10.2.0.1]      [10.2.0.2]     FTP: R PORT=1033   230 User logged in OK - Proceed
15  [10.2.0.2]      [10.2.0.1]     TCP: D=21 S=1033     ACK=10832174 WIN=8618
16  [10.2.0.2]      [10.2.0.1]     FTP: C PORT=21   QUIT
17  [10.2.0.1]      [10.2.0.2]     FTP: R PORT=1033   221 Thank you for visiting my FTP site.
18  [10.2.0.2]      [10.2.0.1]     TCP: D=21 S=1033 FIN ACK=10832216 SEQ=182953160 LEN=0 WIN=8576
19  [10.2.0.1]      [10.2.0.2]     TCP: D=1033 S=21 FIN ACK=182953161 SEQ=10832216 LEN=0 WIN=8729
20  [10.2.0.2]      [10.2.0.1]     TCP: D=21 S=1033     ACK=10832217 WIN=8576
```

Figure 1: *In the Sniffer summary window, Sniffer color codes the packets based on the highest decoded layer.*

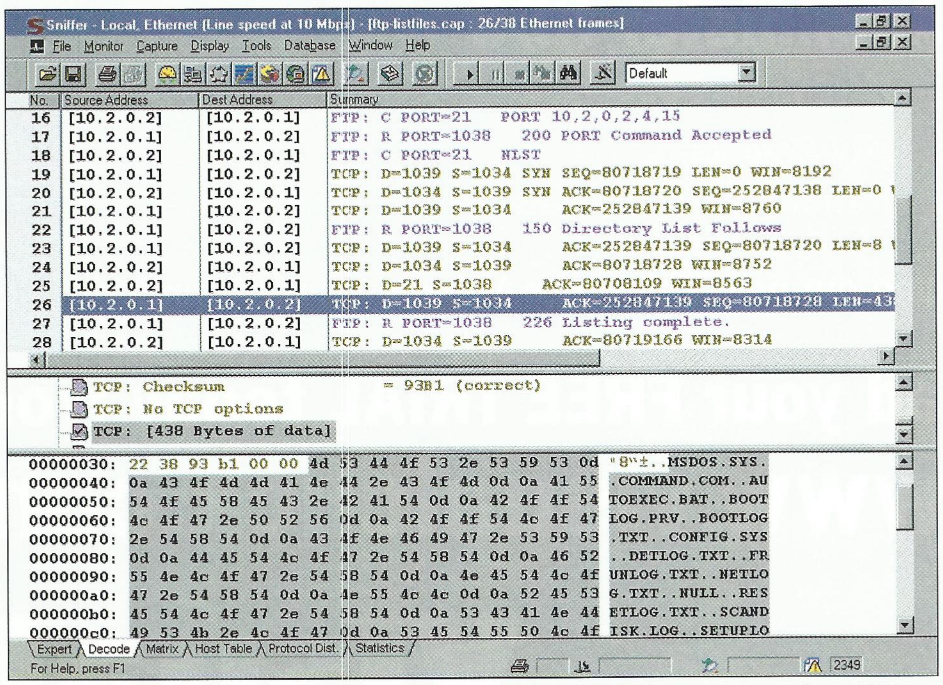
In packets 10 and 13, the user supplies his name (fred) and password (krueger). In packets 11 and 14, the FTP server replies with 331 (User name OK—need password) and 230 (User logged in OK—Proceed).

The most commonly used FTP commands are listed below:

ABOR	abort previous FTP command
LIST and NLST	list files and directories
PASS	send password
PORT	request open port number on specific IP address/port number
QUIT	log off from server
RETR	retrieve file
STOR	send or put file
SYST	identify system type
TYPE	specify file type (A for ASCII, I for Image/binary)
USER	send username

The replies the server may send to these and other FTP commands are listed in "Replies to FTP Commands." (See p. 34.) In addition, some of the fault replies are defined in the "Identifying FTP Faults" section on page 30.

As you can see in Figure 1, the network analyzer should decode the FTP commands and replies. If your network analyzer doesn't decode the FTP commands and replies, you should consider getting a new analyzer.



Sniffer - Local, Ethernet (Line speed at 10 Mbps) - [ftp-listfiles.cap : 26/38 Ethernet frames]
File Monitor Capture Display Tools Database Window Help
Default
Expert Decode Matrix Host Table Protocol Dist. Statistics
For Help, press F1
2349

```
No. Source Address  Dest Address   Summary
16  [10.2.0.2]      [10.2.0.1]     FTP: C PORT=21   PORT 10,2,0,2,4,15
17  [10.2.0.1]      [10.2.0.2]     FTP: R PORT=1038   200 PORT Command Accepted
18  [10.2.0.2]      [10.2.0.1]     FTP: C PORT=21   NLST
19  [10.2.0.1]      [10.2.0.2]     TCP: D=1039 S=1034 SYN SEQ=80718719 LEN=0 WIN=8192
20  [10.2.0.2]      [10.2.0.1]     TCP: D=1034 S=1039 SYN ACK=80718720 SEQ=252847138 LEN=0
21  [10.2.0.1]      [10.2.0.2]     TCP: D=1039 S=1034     ACK=252847139 WIN=8760
22  [10.2.0.1]      [10.2.0.2]     FTP: R PORT=1038   150 Directory List Follows
23  [10.2.0.1]      [10.2.0.2]     TCP: D=1039 S=1034     ACK=252847139 SEQ=80718720 LEN=8
24  [10.2.0.2]      [10.2.0.1]     TCP: D=1034 S=1039     ACK=80718728 WIN=8752
25  [10.2.0.2]      [10.2.0.1]     TCP: D=21 S=1038    ACK=80708109 WIN=8563
26  [10.2.0.1]      [10.2.0.2]     TCP: D=1039 S=1034     ACK=252847139 SEQ=80718728 LEN=43
27  [10.2.0.1]      [10.2.0.2]     FTP: R PORT=1038   226 Listing complete.
28  [10.2.0.2]      [10.2.0.1]     TCP: D=1034 S=1039     ACK=80719166 WIN=8314

    TCP: Checksum              = 93B1 (correct)
    TCP: No TCP options
    TCP: [438 Bytes of data]

00000030: 22 38 93 b1 00 00 4d 53 44 4f 53 2e 53 59 53 0d "8"±..MSDOS.SYS.
00000040: 0a 43 4f 4d 4d 41 4e 44 2e 43 4f 4d 0d 0a 41 55 .COMMAND.COM..AU
00000050: 54 4f 45 58 45 43 2e 42 41 54 0d 0a 42 4f 4f 54 TOEXEC.BAT..BOOT
00000060: 4c 4f 47 2e 50 52 56 0d 0a 42 4f 4f 54 4c 4f 47 LOG.PRV..BOOTLOG
00000070: 2e 54 58 54 0d 0a 43 4f 4e 46 49 47 2e 53 59 53 .TXT..CONFIG.SYS
00000080: 0d 0a 44 45 54 4c 4f 47 2e 54 58 54 0d 0a 46 52 ..DETLOG.TXT..FR
00000090: 55 4e 4c 4f 47 2e 54 58 54 0d 0a 4e 45 54 4c 4f UNLOG.TXT..NETLO
000000a0: 47 2e 54 58 54 0d 0a 4e 55 4c 4c 0d 0a 52 45 53 G.TXT..NULL..RES
000000b0: 45 54 4c 4f 47 2e 54 58 54 0d 0a 53 43 41 4e 44 ETLOG.TXT..SCAND
000000c0: 49 53 4b 2e 4c 4f 47 0d 0a 53 45 54 55 50 4c 4f ISK.LOG..SETUPLO
```

Figure 2: *After an FTP connection is established, most FTP clients request a list of files. The FTP server sends the file list via the newly established connection.*

4. Log out (packets 16-17). When the FTP client logs out (in this case, I typed quit to log out), the FTP server responds with the logout message.
5. Terminate TCP connection (packets 18, 19 and 20). Finally, the FTP client sends the FIN (finish) flag to indicate that it is finished sending data. This is the final step to terminating the TCP connection with the FTP server.

You probably noticed that no files were transferred during this simple sequence. The communication process for transferring files is discussed later in this article.

LISTING FILES ON AN FTP SERVER

After a connection is established, most FTP clients view a list of files on the FTP server. This communication process is shown in Figure 2.

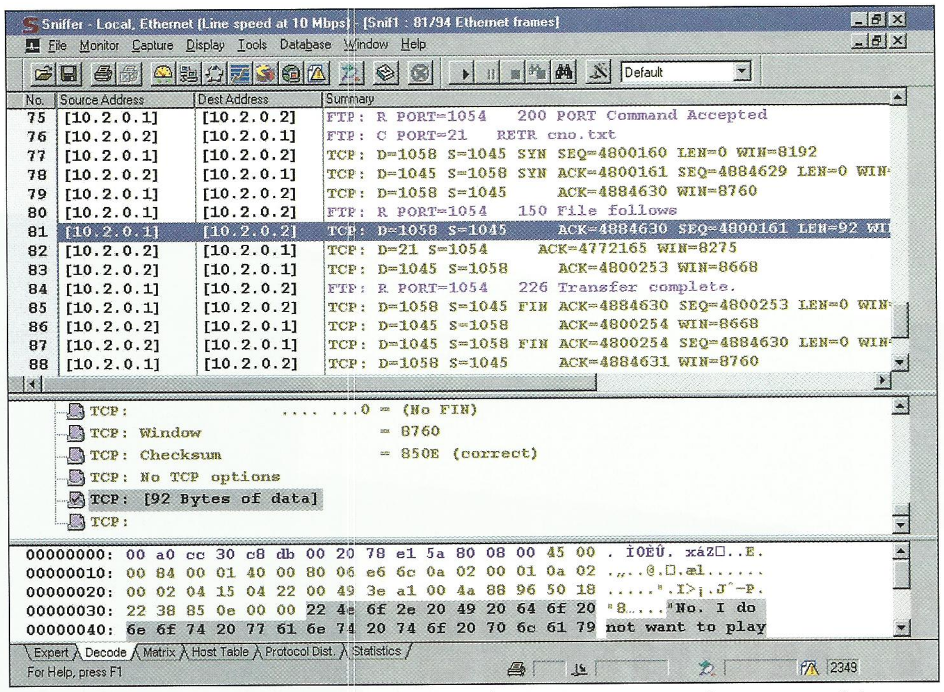
Before receiving the directory list, the FTP client issues a PORT command (PORT 10,2,0,2,4,15). The PORT command tells the server to open a TCP connection to a specific device (the FTP client) using a specific port number. The PORT command can be interpreted as follows:

```
PORT n1.n2.n3.n4,p1,p2
```

In the PORT command, n1.n2.n3.n4 is the FTP client's IP address, and p1 x 256 + p2 is the desired port number. For example, in the command PORT 10,2,0,2,4,15, the FTP client is asking the server to make a connection to the client (10.2.0.2) using the server's source port 1039 (4 x 256 + 15). This connection is then used to transfer the file list.

The actual command to list a directory is the NLST (name list) command (packet 18), which the FTP client sends to the server's FTP port 21. (See Figure 2.) This command triggers the handshake process to open up port 1039, as the FTP client requested, and the directory list reply (packet 26). In packet 26, you will notice that each file in the file list is delimited by the end-of-line sequence 0x0d0a.

Get the Trace! To view the entire trace, you can download the ftp-listfiles.zip file from the Traces section at www.packet-level.com. The ftp-listfiles.zip file contains the ftp-listfiles.cap trace file and the ftp-listfiles.pdf file. As you view the entire trace, you'll notice a second connection

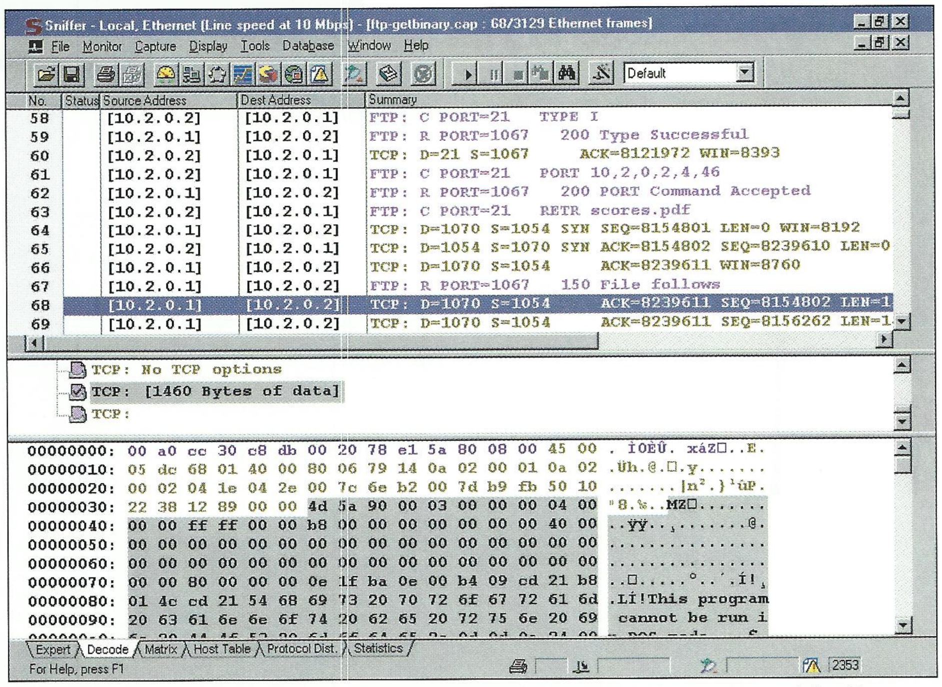

Sniffer - Local, Ethernet (Line speed at 10 Mbps) - [Snif1 : 81/94 Ethernet frames]
File Monitor Capture Display Tools Database Window Help
Default
Expert Decode Matrix Host Table Protocol Dist. Statistics
For Help, press F1
2349

```
No. Source Address  Dest Address    Summary
 75 [10.2.0.1]      [10.2.0.2]      FTP: R PORT=1054   200 PORT Command Accepted
 76 [10.2.0.2]      [10.2.0.1]      FTP: C PORT=21   RETR cno.txt
 77 [10.2.0.1]      [10.2.0.2]      TCP: D=1058 S=1045 SYN SEQ=4800160 LEN=0 WIN=8192
 78 [10.2.0.2]      [10.2.0.1]      TCP: D=1045 S=1058 SYN ACK=4800161 SEQ=4884629 LEN=0 WIN
 79 [10.2.0.1]      [10.2.0.2]      TCP: D=1058 S=1045     ACK=4884630 WIN=8760
 80 [10.2.0.1]      [10.2.0.2]      FTP: R PORT=1054   150 File follows
 81 [10.2.0.1]      [10.2.0.2]      TCP: D=1058 S=1045     ACK=4884630 SEQ=4800161 LEN=92 WI
 82 [10.2.0.2]      [10.2.0.1]      TCP: D=21 S=1054     ACK=4772165 WIN=8275
 83 [10.2.0.2]      [10.2.0.1]      TCP: D=1045 S=1058     ACK=4800253 WIN=8668
 84 [10.2.0.1]      [10.2.0.2]      FTP: R PORT=1054   226 Transfer complete.
 85 [10.2.0.1]      [10.2.0.2]      TCP: D=1058 S=1045 FIN ACK=4884630 SEQ=4800253 LEN=0 WIN
 86 [10.2.0.2]      [10.2.0.1]      TCP: D=1045 S=1058     ACK=4800254 WIN=8668
 87 [10.2.0.2]      [10.2.0.1]      TCP: D=1045 S=1058 FIN ACK=4800254 SEQ=4884630 LEN=0 WIN
 88 [10.2.0.1]      [10.2.0.2]      TCP: D=1058 S=1045     ACK=4884631 WIN=8760

    TCP:                  .... ...0 = (No FIN)
    TCP: Window                     = 8760
    TCP: Checksum                   = 850E (correct)
    TCP: No TCP options
    TCP: [92 Bytes of data]
    TCP:

00000000: 00 a0 cc 30 c8 db 00 20 78 e1 5a 80 08 00 45 00 . ÌOÈÛ. xáZ□..E.
00000010: 00 84 00 01 40 00 80 06 e6 6c 0a 02 00 01 0a 02 .„..@.□.æl......
00000020: 00 02 04 15 04 22 00 49 3e a1 00 4a 88 96 50 18 .....".I>¡.J^-P.
00000030: 22 38 85 0e 00 00 22 4e 6f 2e 20 49 20 64 6f 20 "8....."No. I do
00000040: 6e 6f 74 20 77 61 6e 74 20 74 6f 20 70 6c 61 79 not want to play
```

Figure 3: *If an FTP client sends text files via the FTP connection, the contents of these text files can be read in the hex window of the network analyzer.*

termination sequence for the connection to port 1039. This connection termination process is instigated by the FTP server to indicate that the file list is complete.

Finally, in packet 30, the server sends a TCP packet with the FIN flag set, indicating that the list is complete. (Figure 2 does not show packet 30. You can only view packet 30 in the ftp-listfiles.cap trace file.) If you download and view the trace file on a network analyzer, you will notice minimal time elapses between processes. This is further indication that this process must be automated.


Sniffer - Local, Ethernet (Line speed at 10 Mbps) - [ftp-getbinary.cap : 68/3129 Ethernet frames]
File Monitor Capture Display Tools Database Window Help
Default
Expert Decode Matrix Host Table Protocol Dist. Statistics
For Help, press F1
2353

```
No. Status Source Address  Dest Address    Summary
 58        [10.2.0.2]      [10.2.0.1]      FTP: C PORT=21   TYPE I
 59        [10.2.0.1]      [10.2.0.2]      FTP: R PORT=1067   200 Type Successful
 60        [10.2.0.2]      [10.2.0.1]      TCP: D=21 S=1067     ACK=8121972 WIN=8393
 61        [10.2.0.2]      [10.2.0.1]      FTP: C PORT=21   PORT 10,2,0,2,4,46
 62        [10.2.0.1]      [10.2.0.2]      FTP: R PORT=1067   200 PORT Command Accepted
 63        [10.2.0.2]      [10.2.0.1]      FTP: C PORT=21   RETR scores.pdf
 64        [10.2.0.1]      [10.2.0.2]      TCP: D=1070 S=1054 SYN SEQ=8154801 LEN=0 WIN=8192
 65        [10.2.0.2]      [10.2.0.1]      TCP: D=1054 S=1070 SYN ACK=8154802 SEQ=8239610 LEN=0
 66        [10.2.0.1]      [10.2.0.2]      TCP: D=1070 S=1054     ACK=8239611 WIN=8760
 67        [10.2.0.1]      [10.2.0.2]      FTP: R PORT=1067   150 File follows
 68        [10.2.0.1]      [10.2.0.2]      TCP: D=1070 S=1054     ACK=8239611 SEQ=8154802 LEN=1
 69        [10.2.0.1]      [10.2.0.2]      TCP: D=1070 S=1054     ACK=8239611 SEQ=8156262 LEN=1

    TCP: No TCP options
    TCP: [1460 Bytes of data]
    TCP:

00000000: 00 a0 cc 30 c8 db 00 20 78 e1 5a 80 08 00 45 00 . ÌOÈÛ. xáZ□..E.
00000010: 05 dc 68 01 40 00 80 06 79 14 0a 02 00 01 0a 02 .Üh.@.□.y.......
00000020: 00 02 04 1e 04 2e 00 7c 6e b2 00 7d b9 fb 50 10 .......|n².}¹ûP.
00000030: 22 38 12 89 00 00 4d 5a 90 00 03 00 00 00 04 00 "8.‰..MZ□.......
00000040: 00 00 ff ff 00 00 b8 00 00 00 00 00 00 00 40 00 ..ÿÿ..¸.......@.
00000050: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00000060: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00000070: 00 00 80 00 00 00 0e 1f ba 0e 00 b4 09 cd 21 b8 ..□.....º..´.Í!¸
00000080: 01 4c cd 21 54 68 69 73 20 70 72 6f 67 72 61 6d .LÍ!This program
00000090: 20 63 61 6e 6e 6f 74 20 62 65 20 72 75 6e 20 69  cannot be run i
```

Figure 4. *If an FTP client sends binary files via the FTP connection, most of the file contents are not readable in the hex window. However, some embedded error messages and header information may be contained in binary files.*

Note. If a user is browsing directories and listing their contents, you will find many CWD (change working directory) commands used as well.

COMMANDS FOR TRANSFERRING FILES VIA FTP

The two FTP commands used to retrieve and send files are RETR (retrieve) and STOR (put).

The RETR Command

This section outlines the process of transferring files by explaining the process of transferring the cno.txt file from the remote system. In Figure 3, you can see the RETR command followed by the file name cno.txt (packet 76).

Get the Trace! Download the ftp-getfile.zip file that contains ftp-getfile.cap and ftp-getfile.pdf trace file from the Traces section at www.packet-level.com. Check out how many connections are established before the file transfer occurs.

As you can see in Figure 3, the plain text files are sent across in readable format. You can use network analyzers to capture and view these files as they are being transferred.

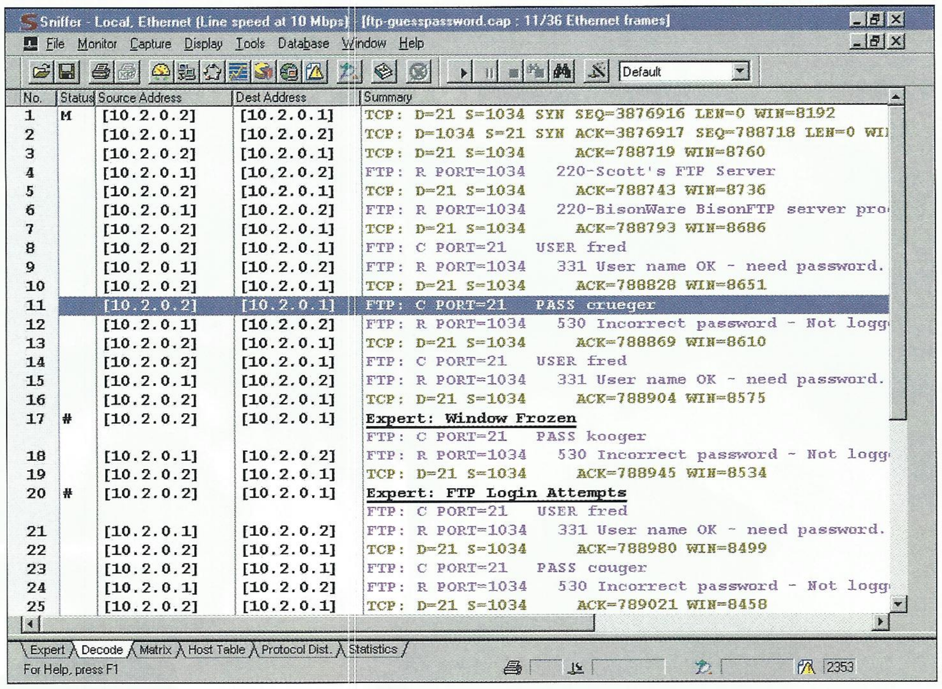
The STOR Process

The STOR process is similar to the RETR process. You can download the ftp-putfile.cap trace file and view the process of moving a file from the FTP client to the server. When you review this trace file, you will notice that I used NLST to recheck the directory contents after I put the file on the FTP server in the eGames directory.

Note. I strongly recommend that you create an RETR or STOR filter for your network analyzer to identify FTP packets being transferred across your company's network.

TRANSFERRING BINARY FILES VIA FTP

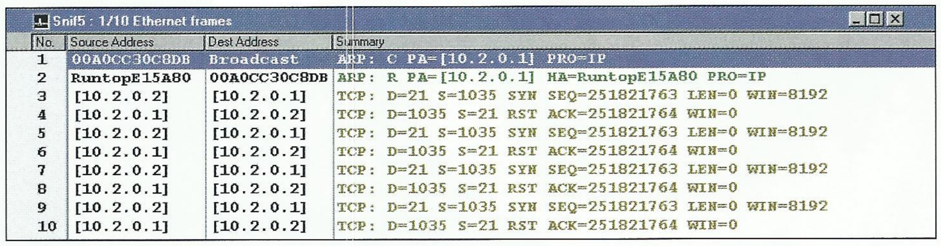
Thus far, the file transfers have involved text files—and, as you can see, these text files are sent across the wire in a readable format. The transfer of binary files, however, requires the FTP client to set a data type of I (to indicate the Image file type). Figure 4 shows the entire file transfer sequence of a large file (more than 2 MB). When I typed the bin command to set up the file transfer for binary mode, the FTP client issued the TYPE I command.



```
Sniffer - Local, Ethernet (Line speed at 10 Mbps) - [ftp-guesspassword.cap : 11/36 Ethernet frames]
No. Status Source Address  Dest Address   Summary
1   M      [10.2.0.2]      [10.2.0.1]     TCP: D=21 S=1034 SYN SEQ=3876916 LEN=0 WIN=8192
2          [10.2.0.1]      [10.2.0.2]     TCP: D=1034 S=21 SYN ACK=3876917 SEQ=788718 LEN=0 WI
3          [10.2.0.2]      [10.2.0.1]     TCP: D=21 S=1034      ACK=788719 WIN=8760
4          [10.2.0.1]      [10.2.0.2]     FTP: R PORT=1034   220-Scott's FTP Server
5          [10.2.0.2]      [10.2.0.1]     TCP: D=21 S=1034      ACK=788743 WIN=8736
6          [10.2.0.1]      [10.2.0.2]     FTP: R PORT=1034   220-BisonWare BisonFTP server pro
7          [10.2.0.2]      [10.2.0.1]     TCP: D=21 S=1034      ACK=788793 WIN=8686
8          [10.2.0.2]      [10.2.0.1]     FTP: C PORT=21   USER fred
9          [10.2.0.1]      [10.2.0.2]     FTP: R PORT=1034   331 User name OK - need password.
10         [10.2.0.2]      [10.2.0.1]     TCP: D=21 S=1034      ACK=788828 WIN=8651
11         [10.2.0.2]      [10.2.0.1]     FTP: C PORT=21   PASS crueger
12         [10.2.0.1]      [10.2.0.2]     FTP: R PORT=1034   530 Incorrect password - Not logg
13         [10.2.0.2]      [10.2.0.1]     TCP: D=21 S=1034      ACK=788869 WIN=8610
14         [10.2.0.2]      [10.2.0.1]     FTP: C PORT=21   USER fred
15         [10.2.0.1]      [10.2.0.2]     FTP: R PORT=1034   331 User name OK - need password.
16         [10.2.0.2]      [10.2.0.1]     TCP: D=21 S=1034      ACK=788904 WIN=8575
17  #      [10.2.0.2]      [10.2.0.1]     Expert: Window Frozen
                                          FTP: C PORT=21   PASS kooger
18         [10.2.0.1]      [10.2.0.2]     FTP: R PORT=1034   530 Incorrect password - Not logg
19         [10.2.0.2]      [10.2.0.1]     TCP: D=21 S=1034      ACK=788945 WIN=8534
20  #      [10.2.0.2]      [10.2.0.1]     Expert: FTP Login Attempts
                                          FTP: C PORT=21   USER fred
21         [10.2.0.1]      [10.2.0.2]     FTP: R PORT=1034   331 User name OK - need password.
22         [10.2.0.2]      [10.2.0.1]     TCP: D=21 S=1034      ACK=788980 WIN=8499
23         [10.2.0.2]      [10.2.0.1]     FTP: C PORT=21   PASS couger
24         [10.2.0.1]      [10.2.0.2]     FTP: R PORT=1034   530 Incorrect password - Not logg
25         [10.2.0.2]      [10.2.0.1]     TCP: D=21 S=1034      ACK=789021 WIN=8458
For Help, press F1                                                                  2353
```

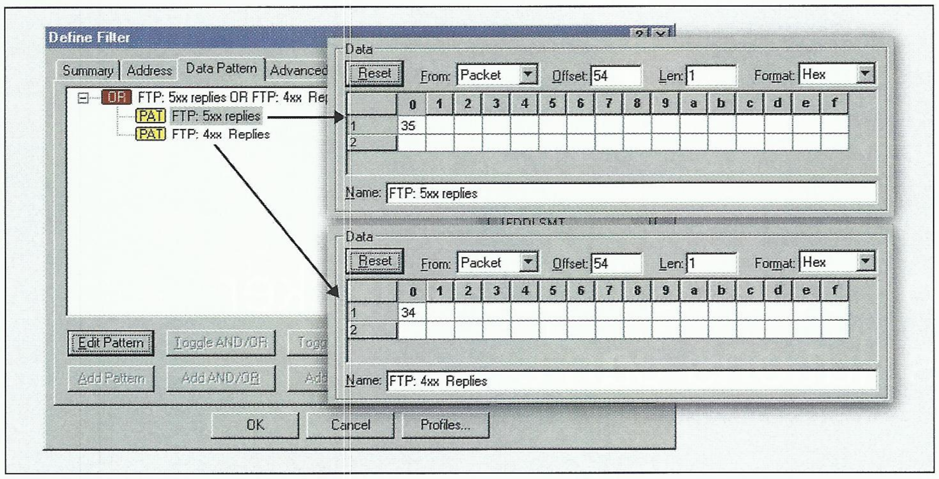
File Monitor Capture Display Tools Database Window Help
Default
Expert Decode Matrix Host Table Protocol Dist. Statistics

Figure 5. *This summary screen shows someone trying to log in by guessing the password for a user account. As you can see, each password attempt is phonetically similar.*



```
Snif5 : 1/10 Ethernet frames
No. Source Address  Dest Address   Summary
1   00A0CC30C8DB    Broadcast      ARP: C PA=[10.2.0.1] PRO=IP
2   RuntopE15A80    00A0CC30C8DB   ARP: R PA=[10.2.0.1] HA=RuntopE15A80 PRO=IP
3   [10.2.0.2]      [10.2.0.1]     TCP: D=21 S=1035 SYN SEQ=251821763 LEN=0 WIN=8192
4   [10.2.0.1]      [10.2.0.2]     TCP: D=1035 S=21 RST ACK=251821764 WIN=0
5   [10.2.0.2]      [10.2.0.1]     TCP: D=21 S=1035 SYN SEQ=251821763 LEN=0 WIN=8192
6   [10.2.0.1]      [10.2.0.2]     TCP: D=1035 S=21 RST ACK=251821764 WIN=0
7   [10.2.0.2]      [10.2.0.1]     TCP: D=21 S=1035 SYN SEQ=251821763 LEN=0 WIN=8192
8   [10.2.0.1]      [10.2.0.2]     TCP: D=1035 S=21 RST ACK=251821764 WIN=0
9   [10.2.0.2]      [10.2.0.1]     TCP: D=21 S=1035 SYN SEQ=251821763 LEN=0 WIN=8192
10  [10.2.0.1]      [10.2.0.2]     TCP: D=1035 S=21 RST ACK=251821764 WIN=0
```

Figure 6. *In this summary screen, the TCP RST (reset) flags indicate that the FTP connection was refused.*



```
Define Filter
Summary | Address | Data Pattern | Advanced
OR  FTP: 5xx replies OR FTP: 4xx Rep
    PAT  FTP: 5xx replies
    PAT  FTP: 4xx Replies

Data
Reset  From: Packet  Offset: 54  Len: 1  Format: Hex
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
1    35
2
Name: FTP: 5xx replies

Data
Reset  From: Packet  Offset: 54  Len: 1  Format: Hex
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
1    34
2
Name: FTP: 4xx Replies
```

Figure 7. *Many network analyzers allow you to build filters on specific data offsets. Boolean filters can use the following three expressions: AND, OR, and NOT. This filter is looking for packets that start with 4 OR 5.*

File transfers are performed directly over TCP in both ASCII mode and image mode.

Get the Trace! To view the summary information of this file transfer, you can download the ftp-getbinary.zip file from the Traces section at www.packet-level.com. The ftp-getbinary.zip file contains the ftp-getbinary.pdf trace file.

IDENTIFYING FTP FAULTS

Interestingly, FTP has a somewhat organized command reply system. In general, replies starting with 4xx are seen as temporary errors. For example, suppose your workstation needs to read a file that is stored on your hard drive to send that file to an FTP server. Because another application has the file open, however, the workstation cannot read the file, and you may receive a 450 FTP reply ("Requested file action not taken; file unavailable").

The replies starting with 500 indicate that a more permanent error has occurred. For example, if the file is not found, you may receive a 550 reply ("Requested action not taken; file unavailable").

You can build a filter for all FTP replies starting with a 4 or a 5. (For more information, see "FTP Filters You Should Build" on p. 32.) The most common FTP errors are listed below:

- **Wrong Username.** If a user enters the wrong username, that user receives a reply 530. Typically, the FTP server also sends a text message stating that the username is unrecognized and that the user is not logged in. These actual text message interpretations are typically self-explanatory.

  Get the Trace! To view the reply 530, download the ftp-wronguser.zip file from the Traces section at www.packet-level.com. The ftp-wronguser.zip file contains the ftp-wronguser.cap trace file.

- **Wrong Password.** When a user tries to make an FTP connection with the wrong password, that user receives reply 530 (just as the wrong username generates). The reply, however, also indicates that the user has supplied an incorrect password and is not logged in.

  In the case of the communications shown in Figure 5, you should be on the alert. This summary screen suggests that someone is trying every variation of a password he or she may have overheard.

Get the Trace! To view the entire trace, you can download the ftp-guesspassword.zip file from the Traces section at www.packet-level.com. The ftp-guesspassword.zip file contains the ftp-guesspassword.cap trace file.

- **FTP Process Is Down: Connection Refused.** The Connection Refused reply is sent when the destination device cannot or will not establish a TCP connection on port 21. The sequence shown in Figure 6 illustrates the problem. (See p. 30.) You may see this problem when an FTP server is down or someone has decided to move the FTP daemon to a nonstandard port number.

Get the Trace! To view the entire trace, you can download the ftp-reuse.zip file from the Traces section at www. packet-level.com. The ftp-reuse.zip file contains the ftp-refuse.cap trace file and the ftp-refuse.pdf file.

Note. Allowing users to activate an FTP server on a nonstandard port number is a huge security risk since most firewalls block FTP traffic based on port 21. If a user has an active FTP server on port 1050, for example, a hacker could easily transfer files, and your company's firewall would not be able to detect and prevent the transfer of these files.

- **User Access Is Limited.** In the case of a file size limitation, the FTP server will notify an FTP client that they have requested a file that is larger than they are allowed to get.

  If a user requests a file that is larger than the allowable file size, that user receives reply 550 ("File size exceeds available credits").

Get the Trace! To view the entire trace showing this error message, you can download the ftp-filesizeprob.zip file from the Traces section at www.packet-level.com. The ftp-filesizeprob.zip file

FTP Filters You Should Build

To capture and examine FTP communications effectively and efficiently, you can build and save the following filters on your network analyzer:

Filter Name	Filtered Field(s)	Value	Definition
FTP Commands	Source Port (offset 34 from start of packet) Destination Port (offset 36 from start of packet)	21 (2 bytes) 21 (2 bytes)	Captures all commands and their respective replies. Good for characterizing FTP traffic and looking for responses indicating faults.
FTP PORT Command	Data portion after the TCP header (offset 54 from start of packet).	0x50-4F-52-54 (4 bytes)	Find out which ephemeral ports are being opened to support information transfer initiated by FTP
FTP NLST (name list) Command	Data portion after the TCP header (offset 54 from start of packet)	0x4E-4C-53-54 (4 bytes)	Find out which users are viewing directory lists. The filter is especially useful if you are concerned about hackers.
FTP CWD (change working directory) Command	Data portion after the TCP header (offset 54 from start of packet)	0x43 57 44 (3 bytes)	Find out which directories users are searching for. This filter is especially useful when you are concerned about hackers. You may want to build a boolean filter that searches for all NLST or CWD activity.
FTP RETR (retrieve) Command	Data portion after the TCP header (offset 54 from start of packet)	0x52 45 54 52 (4 bytes)	Identify the files that are being transferred across the wire. Ideally, this filter should be combined with a STOR filter. Using a combined boolean filter of RETR or STOR activity, you can identify the source and destination of all files transferred using FTP. **Note.** This filter works even if an ephemeral port is being used because it looks for a specific byte pattern up in the data portion of the packet.
FTP Negative Replies	Data portion after the TCP header (offset 54 from start of packet)	0x34 or 0x35 (1 byte each).	Use this filter to classify the typical FTP errors that occur on a network. A high number of errors indicates that an FTP client is not getting what it wants. The FTP client may be overstepping its boundaries, or the FTP server may be overly protective—that's your call. Check it out.

Note. These filters assume that the Ethernet II frame type is in use. (Figure 7on p. 30 shows the breakdown of a boolean filter looking for FTP negative replies that start with 4 or 5 (in ASCII as translated to hexadecimal). ●

Replies to FTP Commands

As you decode FTP communications, you should know the possible replies to FTP commands. The following replies are defined in Request For Comments (RFC) 959 in section 4.2.2. (You can download this RFC from www.ietf.org/rfc.)

100 REPLIES: POSITIVE PRELIMINARY REPLY
110 Restart marker reply.
120 Service ready in *nnn* minutes.
125 Data connection already open; transfer starting.
150 File status okay; about to open data connection.

200 REPLIES: POSITIVE COMPLETION REPLY
200 Command okay.
202 Command not implemented, superfluous at this site.
211 System status, or system help reply.
212 Directory status.
213 File status.
214 Help message.
215 NAME system type.
220 Service ready for new user.
221 Service closing control connection; logged out if appropriate.
225 Data connection open; no transfer in progress.
226 Closing data connection; requested file action successful (for example, file transfer or file abort).
227 Entering Passive Mode (h1,h2,h3,h4,p1,p2).
230 User logged in, proceed.
250 Requested file action okay, completed.
257 "PATHNAME" created.

300 REPLIES: POSITIVE INTERMEDIATE REPLY
331 User name okay, need password.
332 Need account for login.
350 Requested file action pending further information.

400 REPLIES: TRANSIENT NEGATIVE COMPLETION REPLY
421 Service not available, closing control connection; this may be a reply to any command if the service knows it must shut down.
425 Can't open data connection.
426 Connection closed; transfer aborted.
450 Requested file action not taken; file unavailable (for example, file busy).
451 Requested action aborted: local error in processing.
452 Requested action not taken; insufficient storage space

500 REPLIES: PERMANENT NEGATIVE COMPLETION REPLY
500 Syntax error, command unrecognized; this may include errors such as command line too long.
501 Syntax error in parameters or arguments.
502 Command not implemented.
503 Bad sequence of commands.
504 Command not implemented for that parameter.
530 Not logged in.
532 Need account for storing files.
550 Requested action not taken; file unavailable (for example, file not found, no access).
551 Requested action aborted: page type unknown.
552 Requested file action aborted; exceeded storage allocation (for current directory or dataset).
553 Requested action not taken; file name not allowed.

contains the ftp-filesizeprob.cap trace file and the ftp-filesizeprob.pdf file.

TRACING FTP COMMUNICATIONS

I strongly recommend that you start tracing FTP communications on your company's network. I also recommend that you purchase W. Richard Steven's book about TCP/IP and the FTP RFC (listed in the references section at the end of this article). You should also download the trace files mentioned in this article and then start capturing FTP traffic on your company's network.

When you analyze FTP traffic, you should use the following questions to pinpoint problem areas on the network:

- Are the FTP connections successful? Do the handshakes work properly?
- What errors, if any, do you see? Are there any common errors that may indicate a reconfiguration is necessary?
- Is your company's network "ripe" for hacking? If you port scan the network, how many devices come up with port 21 listening?
- Is your company's network being hacked? Build the filters listed in "FTP Filters You Should Build" (see p. 32) and look for a high number of CWD commands on the wire: Is someone snooping around your company's network? If you allow FTP connections from the Internet, consider building a filter for all CWD commands not issued from your internal network addresses.
- Who is doing all this FTP work? Should you be using a more robust file transfer system?
- Are any FTP commands being sent on any ports other than port 21? If so, why? This could thwart your firewall's attempt to block FTP traffic based on the port 21.

Spend some time tracing various FTP sequences on your working network or in a lab environment. When something goes wrong, you can not only locate the problem at the packet-level, but you can also document it for others to review. Watching FTP at the packet-level certainly takes the guesswork out of troubleshooting!

REFERENCES

For more information about FTP, you can read the following:

- Stevens, W. Richard. "TCP/IP Illustrated, Volume I: The Protocols." Addison-Wesley Professional Computing Series
- RFC 959, "File Transfer Protocol (FTP)," October 1985, Postel and Reynolds. Available at www.ietf.org.
- PAIN; the Protocol Analysis Institute network, A Study in Lousy Network Components and Protocols. You can find out more about PAIN at www.packet-level.com.

Laura Chappell is the Senior Protocol Analyst with the Protocol Analysis Institute (www.packet-level.com). She is also the author of several self-paced courses on network analysis, troubleshooting, and Cyber Crime that are available online at www.podbooks.com. Ms. Chappell is a member of IEEE, Usenix, and HTCIA. She will be presenting three analysis seminars at Salford University on October 3–5 (see www.brainstorm-series.com).