PROTOCOL ANALYSIS INSTITUTE

Understanding the FTP PORT Command

Laura Chappell, Sr. Protocol Analyst
Protocol Analysis Institute
www.packet-level.com; www.podbooks.com

Note: HP Certified Professionals may attend a free course online "**Detecting and Preventing Network Scans - Part 1 and Part 2; approx. time: 90 mins. total**" at ftp://ftp.hp.com/pub/hpcp/chappel.html.

===

You may already know that when FTP (File Transfer Protocol) commands cross the wire, they use port 21 by default. You may also know that port 20 is assigned to FTP data. Unfortunately, most FTP data sessions do not actually use port 20.

So you have just taken a trace of an FTP session and noticed that a PORT command crossed the wire. When you looked at the decode, you saw the strangest command sequence:

```
PORT 10,2,0,2,4,31
```

[We have several FTP trace files online at http://www.packet-level.com/traceFiles.htm.]

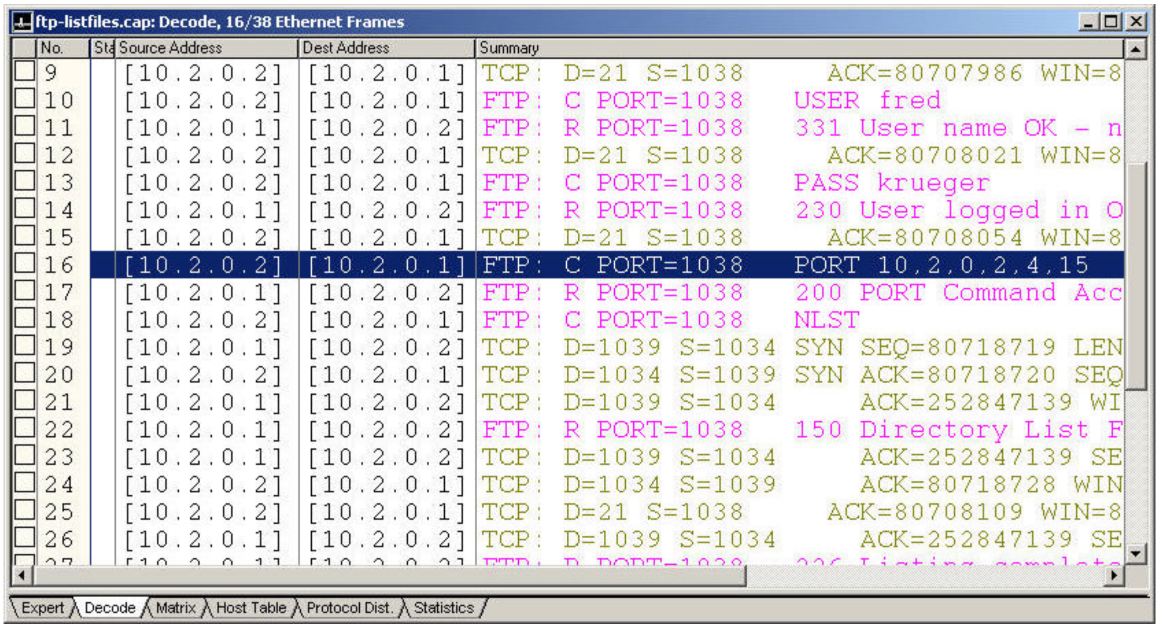
What does this mean? First let us take a look at the purpose of the PORT command. Then we will decipher the numbers following the command.

THE PORT COMMAND

FTP communications use two port number values – one for commands (port 21 by default) and one for data transfer (this is where the PORT command comes into play).

The PORT command is sent by an FTP client to establish a secondary connection (address and port) for data to travel over. In some FTP implementations port 20 is used for data, but that is the exception rather than the rules. Typically in a trace you will see data crossing over a dynamic port number (IANA states that this range should be between 49152 through 65535, but most likely you'll see your application using something just above 1024 – the area that used to be the dynamic port number area).

Figure 1 shows the summary of an FTP communication. Packet 16 contains the PORT command. [This trace file is online at http://www.packet-level.com/traceFiles.htm.]



No.	Sta	Source Address	Dest Address	Summary
□ 9		[10.2.0.2]	[10.2.0.1]	TCP: D=21 S=1038 ACK=80707986 WIN=8
□ 10		[10.2.0.2]	[10.2.0.1]	FTP: C PORT=1038 USER fred
□ 11		[10.2.0.1]	[10.2.0.2]	FTP: R PORT=1038 331 User name OK - n
□ 12		[10.2.0.2]	[10.2.0.1]	TCP: D=21 S=1038 ACK=80708021 WIN=8
□ 13		[10.2.0.2]	[10.2.0.1]	FTP: C PORT=1038 PASS krueger
□ 14		[10.2.0.1]	[10.2.0.2]	FTP: R PORT=1038 230 User logged in O
□ 15		[10.2.0.2]	[10.2.0.1]	TCP: D=21 S=1038 ACK=80708054 WIN=8
□ 16		[10.2.0.2]	[10.2.0.1]	FTP: C PORT=1038 PORT 10,2,0,2,4,15
□ 17		[10.2.0.1]	[10.2.0.2]	FTP: R PORT=1038 200 PORT Command Acc
□ 18		[10.2.0.2]	[10.2.0.1]	FTP: C PORT=1038 NLST
□ 19		[10.2.0.1]	[10.2.0.2]	TCP: D=1039 S=1034 SYN SEQ=80718719 LEN
□ 20		[10.2.0.2]	[10.2.0.1]	TCP: D=1034 S=1039 SYN ACK=80718720 SEQ
□ 21		[10.2.0.1]	[10.2.0.2]	TCP: D=1039 S=1034 ACK=252847139 WI
□ 22		[10.2.0.1]	[10.2.0.2]	FTP: R PORT=1038 150 Directory List F
□ 23		[10.2.0.1]	[10.2.0.2]	TCP: D=1039 S=1034 ACK=252847139 SE
□ 24		[10.2.0.2]	[10.2.0.1]	TCP: D=1034 S=1039 ACK=80718728 WIN
□ 25		[10.2.0.2]	[10.2.0.1]	TCP: D=21 S=1038 ACK=80708109 WIN=8
□ 26		[10.2.0.1]	[10.2.0.2]	TCP: D=1039 S=1034 ACK=252847139 SE
□ 27		[10.2.0.1]	[10.2.0.2]	FTP: R PORT=1038 226 Listing complete

Figure 1: The PORT command and parameters are visible in Sniffer's summary column.

An FTP client issues a PORT to the FTP server and defines what port the client will be listening on for the data channel connection. Upon receipt of the PORT command, the server establishes a new TCP connection to the client using that TCP port value.

You may see numerous PORT commands issued during a single FTP session – a new data channel must be established to transfer directory listings and perform file GET and PUT operations.

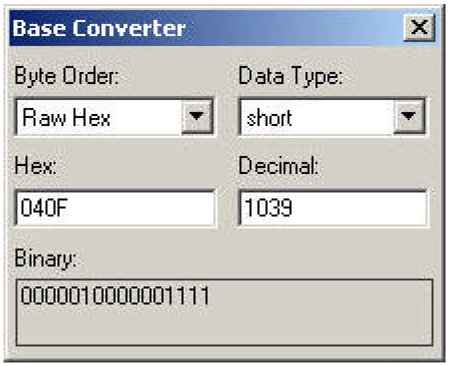
THE FREAKY NUMBERS

After the PORT command, you will see a series of six numbers – these numbers indicate the IP address and port number to use in establishing a data transfer connection. The first four numbers (10,2,0,2 in our example above) indicate the client IP address. The second numbers, 4,15 indicate the client port number.

4,15? Strange. When you look at your trace, you would notice that the server establishes a connection on the client port 1039 (D=1039 in packet 19 in Figure 1). How did we get from 4,15 to 1039? Here we go. To interpret and translate the value 4,15 into a port number the receiver must do some decimal to hex translations – here is an example:

> first number (4) translate to hex (0x04)
> second number (15) translate to hex (0x0F)

Now take the entire set of hex bytes (0x040F) and translate the bytes from hex to decimal (1055). Figure 2 displays the conversion value in Hex Workshop's Base Converter applet. (Hex Workshop and Base Converter are available online at www.bpsoft.com.) Voila!



Base Converter

Byte Order: Raw Hex

Data Type: short

Hex: 040F

Decimal: 1039

Binary: 0000010000001111

Figure 2: Hex value 040F is equivalent to decimal value 1039.

Most folks get snagged when they try to translate both decimal values as a single set (415 = 0x019F) – that just will not work. You must split the values and convert individually to hex before combining and converting to decimal.

Now you know – when you see another PORT command on the wire, you should be able to guess what port the data transfer process will use.

Join the Packeteer Mailing list online at www.packet-level.com for the latest news in analysis, troubleshooting, optimization and security. (Copyright 2003 Protocol Analysis Institute)