EXHIBIT V

Case No. 1:14-cv-00857-TSC-DAR

Understanding the FTP Protocol

by **Don Parker** [Published on **8 Sept. 2005** / Last Updated on **8 Sept. 2005**]

Out of the many protocols in existence today only a couple have been written for the purpose of data transfer. After all, not all of the Internet's activity revolves around HTTP and web pages. This article will cover the FTP protocol and how it goes about actually doing your data transfers for you.

FTP the protocol

Commonly when one thinks of the Internet, the first thing that comes to mind is "surfing" from one website to another. Being able to go from website to another, and view the contents is indeed the reason that the Internet is as popular as it is today, and growing bigger every year. If we set web surfing aside though, just what do we have left in terms of actual usage going on whilst on the Internet? Well one of the activities that takes place is the downloading of data files, movies, anti-virus updates, and the such. What these acts have in common is one protocol, namely the FTP protocol or File Transfer Protocol.

It should be noted that FTP also observes the client/server model. Unlike HTTP though, where there is a clear cut winner for web browsers and web servers, no such program can make the same claim as it relates to FTP. There is a large selection of FTP clients and servers out there today. It is worth noting that your version of Windows come with a built-in FTP client.

FTP itself uses the TCP transport protocol exclusively, or in other words, it never uses UDP for its transport needs. Typically an application layer protocol will use one or the other. One notable exception to that is DNS (http://www.tech-faq.com/dns.shtml) or Domain Name System. FTP also is odd in the fact that it uses two ports to accomplish its task. It typically uses port 20 for data transfer and port 21 to listen to commands. Though having data transferred over port 20 is not always the case as it can also be a different port as well. That is where the confusing part for many people comes into play. There are two modes to FTP, namely active and passive mode. These two modes are initiated by the FTP client, and then acted upon by the FTP server.

Let's delve deeper

So just how does active and passive FTP work anyways? Well it all starts with the FTP client initiating a connection with the FTP server on its port 21. Port 21 is where the server is listening for commands issued to it, and in turn, which it will respond to. So we will assume that the TCP/IP handshake is complete, and as normal the client has done all of this on an ephemeral port (http://www.ncftp.com/ncftpd/doc/misc/ephemeral_ports.html). At this point the client begins to listen on it's ephemeral port + 1, and sends the PORT N+1 command to the server on its port 21 i.e. if the ephemeral port in use by the client is 1026, then it would listen on port 1027. Once this is done the data transfer port (port 20) on the FTP server would initiate a connection to the FTP client's ephemeral port plus 1, as indicated above. This is pretty much how an active FTP session is conducted by both the client and server. Though if the client has a firewall in place, this whole communication process will come to a grinding halt. The clients firewall would drop what it considers to be an unsolicited communication attempt on the ephemeral port plus one port for the data transfer. The way that FTP gets around this problem is by using passive FTP.

Let's take the passive approach

By using the passive mode of FTP or as it appears in the ASCII content of a packet "PASV", FTP was able to neatly sidestep the firewall issue on the client side. It was done in the following fashion: The FTP client, let's say the built in FTP client that comes with a win32 operating system, will start up two connections to the FTP server. We need to keep in mind as well that both connections that are initiated by the client are using ephemeral ports themselves, as it should be. By opening two connections, or sockets with the FTP server, the client is able to resolve the issue of its firewall denying access to the FTP server initiating contact on one of the client's high ephemeral ports.

One of the connections opened by the client will contact the FTP server on port 21, and issue it the PASV (passive) command, vice the normal PORT command when using active FTP. Now what happens is that the FTP server opens an ephemeral port and issues the PORT command to the FTP client. With this in hand the client then starts a connection back to the server port for the data transfer. It is a rather nifty way to deal with the aforementioned issue of Active FTP and client firewalls.

Yet more details

Much like some other application layer protocols, FTP has its own set of status and error codes (http://www.leaning-net.co.uk/content/ftperr.htm). Just like HTTP, these numerical values will tell you what is going on with an established session. Also much like HTTP status and error codes they are broken down into five groups. It is always handy to have a link to a breakout of these nearby if you are investigating some traffic issues. Well with that said, what would an article about a protocol be without a packet showing it in action! Without further ado let's take a look at one of them.

Behold the FTP packet!

```
14:01:25.561863 192.168.1.100.21 > 192.168.1.200.11191: P [tcp sum ok] 825:884(59) ack 367 win 65169 (DF) (ttl 128, id 64059, len 99)
0x0000  4500 0063 ee83 4000 7606 c5e7 c0a8 0164      E..c..@.v.......
0x0010  c0a8 01c8 0015 2bb7 8941 e301 1dc0 b76c      ......+..A.....l
0x0020  5018 fe91 7d81 0000 3135 3020 4f70 656e      P...}...150.Open
0x0030  696e 6720 4249 4e41 5259 206d 6f64 6520      ing.BINARY.mode.
0x0040  6461 7461 2063 6f6e 6e65 6374 696f 6e20      data.connection.
0x0050  666f 7220 4a72 412e 3139 3939 2e6a 7067      for.JrA.1999.jpg
0x0060  2e0d 0a                                      ...
```

So what can we glean from looking at the packet above? Well we can see that the FTP server is located on 192.168.1.100 on port 21, and going to binary mode in order to facilitate the transfer of a picture to the client at 192.168.1.200 on its port 11191. What else can we extract information wise from this packet? If we look at the ttl value as seen above we can reasonably say that the FTP server is a win32 operating system, as the default ttl values on win32 operating systems are 128, and the fact that the DF bit is set pretty much seals the deal as to this being a win32. Pretty neat! There is a lot of information contained in a packet as you can see.

Conclusion

Now for a quick refresher on what protocol starts where! We know that the TCP header starts at bytes 0015 as underlined above, so we can infer from that the IP header ended at the bytes 01c8 preceding bytes 0015. Also seeing as there are no TCP options set as evidenced by the value of 5 (which is underlined in the above packet) we know where the TCP header ends and the FTP data begins. So we can see from the underlined portions in the above packet just where our FTP traffic is. Well on that note I will wrap up this article on FTP, and should you have any questions on this please feel free to drop me a line. Till next time!

See Also

- OSI Reference Model: Layer 7 Hardware (http://www.windowsnetworking.com/articles-tutorials/common/OSI-Reference-Model-Layer7-Hardware.html) **on 21 Oct. 2008 (2008-10-21 11:00) by Russell Hitchcock**
- WPUT FTP upload utility (http://www.windowsnetworking.com/blogs/wnadmin/networking-central/wput-ftp-upload-utility-73.html) **on 23 Oct. 2006 (2006-10-23 10:32) by Vitaly Popovich**
- IIS 7.0 - FTP Publishing Service – Part 2: Configuration (http://www.windowsnetworking.com/articles-tutorials/windows-server-2008/IIS-FTP-Publishing-Service-Part2.html) **on 3 July 2008 (2008-07-03 10:00) by Peter Schmidt**
- FTP Resources (http://www.windowsnetworking.com/kbase/WindowsTips/WindowsNT/AdminTips/Network/FTPResources.html) **on 20 April 2004 (2004-04-20 02:00) by Wayne Maples**
- Isolexis Monitor (http://www.windowsnetworking.com/software/Network-monitoring-management/Isolexis-Monitor.html) **on 5 Jan. 2003 (2003-01-05 01:00) by C.C.T.S. Enterprises**
- sMonitor (http://www.windowsnetworking.com/software/Network-monitoring-management/sMonitor.html) **on 29 July 2003 (2003-07-29 02:00) by Alexander Yarovy**
- Configuring IIS To Host an FTP Site (Part 2) (http://www.windowsnetworking.com/articles-tutorials/network-protocols/Configuring-IIS-Host-FTP-Site-Part2.html) **on 9 April 2009 (2009-04-09 08:00) by Brien M. Posey**
- Backup4all (http://www.windowsnetworking.com/software/Backup-software/Backup4all.html) **on 7 July 2011 (2011-07-07 02:00) by Softland**
- Creating and Configuring FTP Sites in Windows Server 2003 (http://www.windowsnetworking.com/articles-tutorials/windows-2003/Creating-Configuring-FTP.html) **on 11 Aug. 2004 (2004-08-11 17:49) by Mitch Tulloch**
- Planex NAS-01G (http://www.windowsnetworking.com/blogs/wnadmin/news/planex-nas-01g-218.html) **on 9 Feb. 2007 (2007-02-09 06:56) by Vitaly Popovich**