## FTP traffic and capture with Wireshark

By Matthew Miller

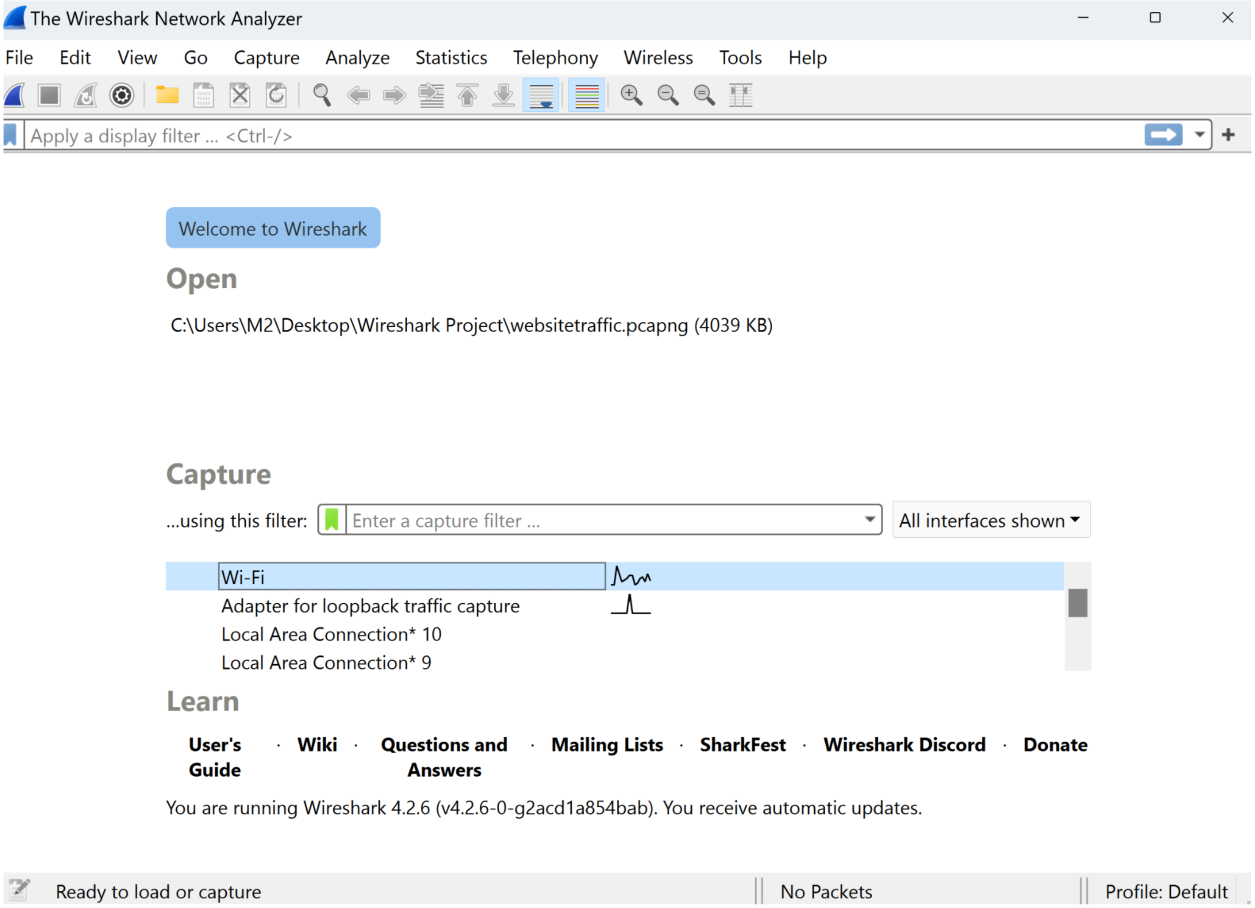
### Introduction

In this project, I will demonstrate how to use Wireshark on a Windows machine to capture and analyze File Transfer Protocol (FTP) traffic. FTP is a standard network protocol used for transferring files between a client and a server on a computer network. Understanding FTP traffic is crucial because it often transmits data, including login credentials, in plain text, making it vulnerable to interception and attacks. By analyzing FTP traffic, I can identify these vulnerabilities and understand the importance of using secure alternatives like SFTP.

### Step 1: Setting Up Wireshark

First, I need to set up Wireshark, a powerful network protocol analyzer that allows me to capture and inspect the data traveling through my network in real-time.

1. **Open Wireshark**: I launch Wireshark by clicking on the Wireshark icon on my desktop or finding it in the Start menu.
2. **Select the Network Interface**: Once Wireshark is open, I select the network interface that I am using to connect to the internet. On a Windows machine, this is usually my Wi-Fi or Ethernet adapter. This step is crucial because selecting the correct interface ensures that I capture the right network traffic.



The Wireshark Network Analyzer
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>
Welcome to Wireshark

Open

C:\Users\M2\Desktop\Wireshark Project\websitetraffic.pcapng (4039 KB)

Capture

...using this filter: Enter a capture filter … All interfaces shown

Wi-Fi
Adapter for loopback traffic capture
Local Area Connection* 10
Local Area Connection* 9

Learn

User's Guide · Wiki · Questions and Answers · Mailing Lists · SharkFest · Wireshark Discord · Donate

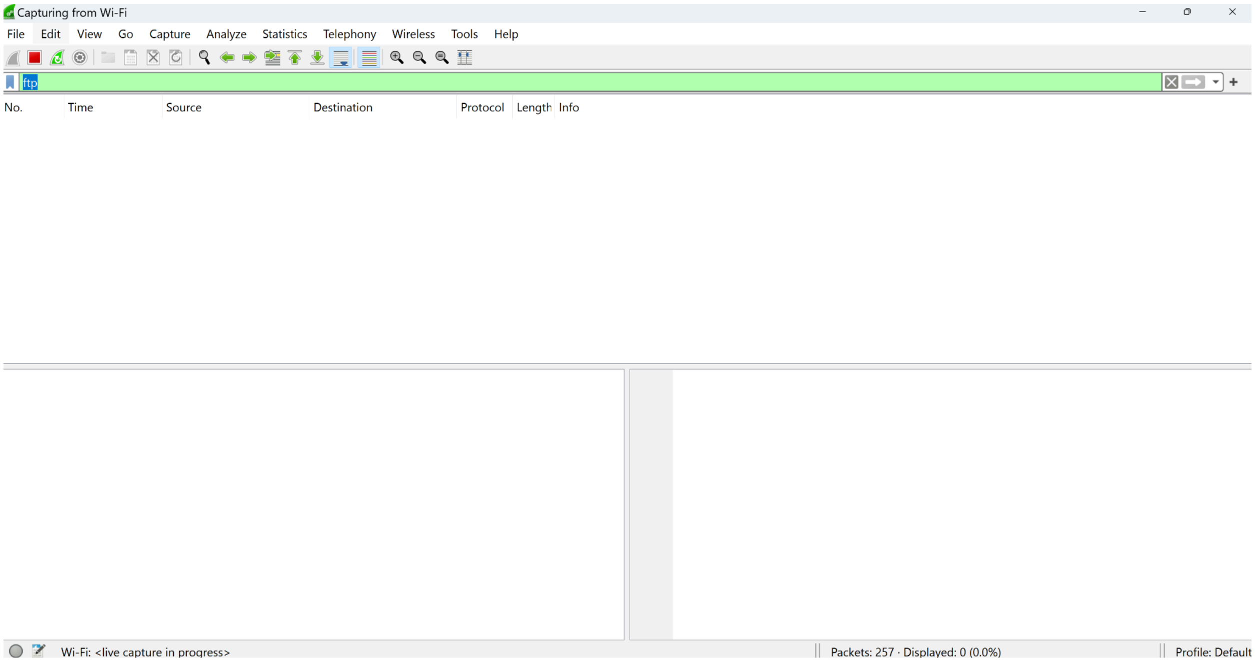
You are running Wireshark 4.2.6 (v4.2.6-0-g2acd1a854bab). You receive automatic updates.

Ready to load or capture | No Packets | Profile: Default

### Step 2: Starting the Capture

Now that I have selected the correct network interface, I begin capturing network traffic.

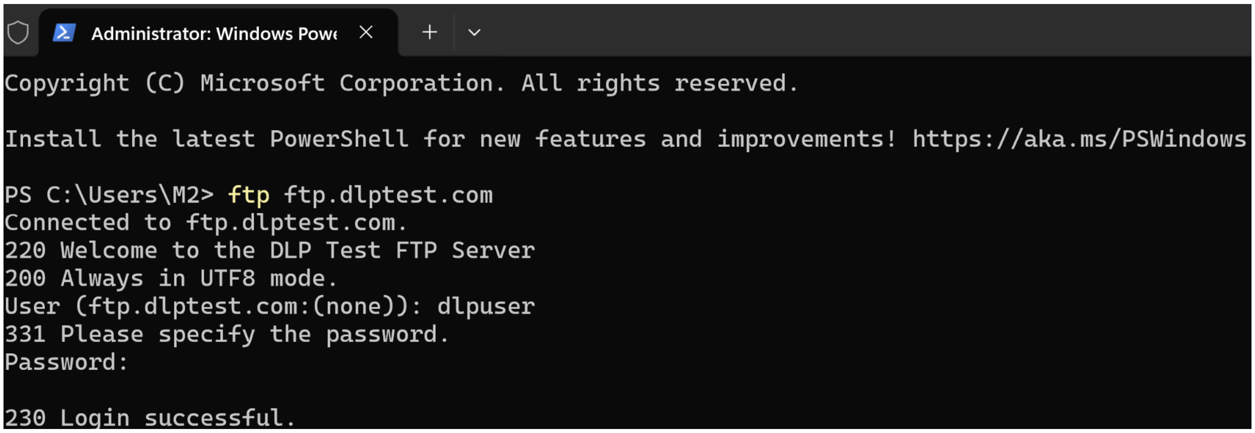
1. **Start Capture**: I click the blue shark fin icon at the top of the Wireshark interface to start capturing packets. Wireshark will now record all the traffic on my selected network interface.
2. **Filter for FTP Traffic**: Since I am specifically interested in FTP traffic, I apply a filter to focus on FTP-related packets. I enter `ftp` in the filter bar at the top of the screen and press Enter. This filter will show only the packets related to the FTP protocol.


Capturing from Wi-Fi
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
ftp
No. Time Source Destination Protocol Length Info
Wi-Fi: <live capture in progress>
Packets: 257 · Displayed: 0 (0.0%)
Profile: Default

### Step 3: Generating FTP Traffic

To analyze FTP traffic, I need to generate some by connecting to an FTP server.

1. **Open Command Prompt**: On my Windows machine, I open the Command Prompt by typing `cmd` in the Start menu search bar and pressing Enter.
2. **Connecting to an FTP Server**: In the Command Prompt, I use an FTP client to connect to an FTP server. For this demonstration, I use the following command: ftp ftp.dlptest.com.
3. **Logging In**: The server will prompt me for a username and password. I enter `dlpuser` as the username and `rNrKYTX9g7z3RgJRmxWuGHbeu` as the password. This information is sent over the network and can be captured by Wireshark.
4. I type ‘bye’ to exit the ftp server.

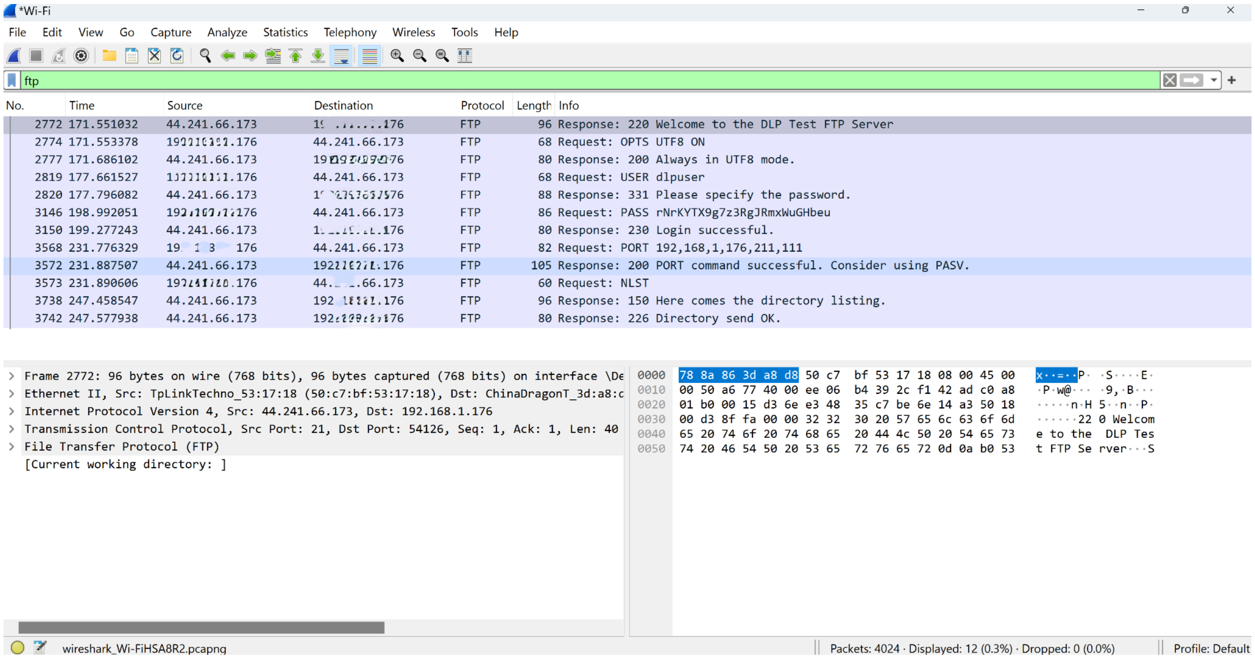


```
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\M2> ftp ftp.dlptest.com
Connected to ftp.dlptest.com.
220 Welcome to the DLP Test FTP Server
200 Always in UTF8 mode.
User (ftp.dlptest.com:(none)): dlpuser
331 Please specify the password.
Password:

230 Login successful.
```

### Step 4: Capturing and Analyzing FTP Traffic

With the connection established, I can now analyze the FTP traffic in Wireshark.

1. **Stop the Capture**: Once I have generated enough FTP traffic, I return to Wireshark and click the red square icon at the top to stop the packet capture.
2. **Inspecting the Packets**: I begin analyzing the captured packets. The list of packets in the top pane shows various details, such as the source and destination IP addresses, the protocol used (FTP), and a brief description of each packet.


*Wi-Fi
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

ftp

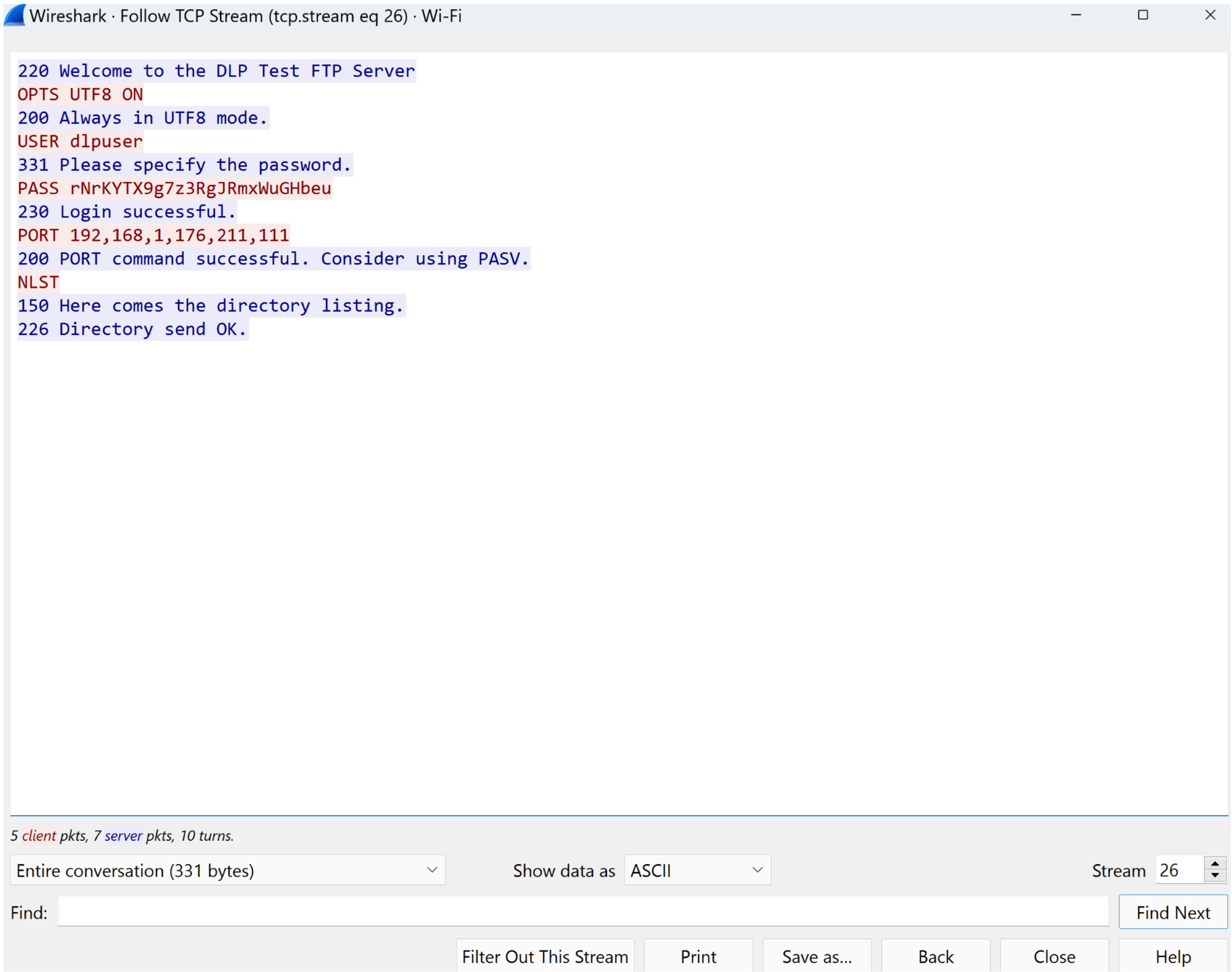
```
No.  Time       Source          Destination     Protocol Length Info
2772 171.551032 44.241.66.173   [redacted].176  FTP       96 Response: 220 Welcome to the DLP Test FTP Server
2774 171.553378 [redacted].176  44.241.66.173   FTP       68 Request: OPTS UTF8 ON
2777 171.686102 44.241.66.173   [redacted]76    FTP       80 Response: 200 Always in UTF8 mode.
2819 177.661527 [redacted].176  44.241.66.173   FTP       68 Request: USER dlpuser
2820 177.796082 44.241.66.173   [redacted]76    FTP       88 Response: 331 Please specify the password.
3146 198.992051 [redacted]176   44.241.66.173   FTP       86 Request: PASS rNrKYTX9g7z3RgJRmxWuGHbeu
3150 199.277243 44.241.66.173   [redacted].176  FTP       80 Response: 230 Login successful.
3568 231.776329 [redacted]176   44.241.66.173   FTP       82 Request: PORT 192,168,1,176,211,111
3572 231.887507 44.241.66.173   [redacted].176  FTP      105 Response: 200 PORT command successful. Consider using PASV.
3573 231.890606 [redacted].176  44.[redacted].66.173 FTP  60 Request: NLST
3738 247.458547 44.241.66.173   [redacted].176  FTP       96 Response: 150 Here comes the directory listing.
3742 247.577938 44.241.66.173   [redacted]176   FTP       80 Response: 226 Directory send OK.
```

```
> Frame 2772: 96 bytes on wire (768 bits), 96 bytes captured (768 bits) on interface \De
> Ethernet II, Src: TpLinkTechno_53:17:18 (50:c7:bf:53:17:18), Dst: ChinaDragonT_3d:a8:c
> Internet Protocol Version 4, Src: 44.241.66.173, Dst: 192.168.1.176
> Transmission Control Protocol, Src Port: 21, Dst Port: 54126, Seq: 1, Ack: 1, Len: 40
> File Transfer Protocol (FTP)
  [Current working directory: ]
```

```
0000  78 8a 86 3d a8 d8 50 c7  bf 53 17 18 08 00 45 00   x··=··P· ·S····E·
0010  00 50 a6 77 40 00 ee 06  b4 39 2c f1 42 ad c0 a8   ·P·w@··· ·9,·B···
0020  01 b0 00 15 d3 6e e3 48  35 c7 be 6e 14 a3 50 18   ·····n·H 5··n··P·
0030  00 d3 8f fa 00 00 32 32  30 20 57 65 6c 63 6f 6d   ······22 0 Welcom
0040  65 20 74 6f 20 74 68 65  20 44 4c 50 20 54 65 73   e to the  DLP Tes
0050  74 20 46 54 50 20 53 65  72 76 65 72 0d 0a b0 53   t FTP Se rver···S
```

wireshark_Wi-FiHSA8R2.pcapng | Packets: 4024 · Displayed: 12 (0.3%) · Dropped: 0 (0.0%) | Profile: Default

3. **Following the FTP Stream**: To see the conversation between the FTP client and server, I right-click on one of the FTP packets and select “Follow” > “TCP Stream.” This opens a new window displaying the entire conversation, including the commands sent by the client and the responses from the server.



Wireshark · Follow TCP Stream (tcp.stream eq 26) · Wi-Fi

```
220 Welcome to the DLP Test FTP Server
OPTS UTF8 ON
200 Always in UTF8 mode.
USER dlpuser
331 Please specify the password.
PASS rNrKYTX9g7z3RgJRmxWuGHbeu
230 Login successful.
PORT 192,168,1,176,211,111
200 PORT command successful. Consider using PASV.
NLST
150 Here comes the directory listing.
226 Directory send OK.
```

*5 client pkts, 7 server pkts, 10 turns.*

Entire conversation (331 bytes)

Show data as ASCII

Stream 26

Find:

  - **Why This Is Important**: By following the stream, I can see the exact commands and responses between the client and server. This includes sensitive information like login credentials, which in FTP are sent in plain text. This highlights a significant security risk, as anyone capturing this traffic can easily read these credentials.

4. **Examining FTP Commands**: I scroll through the TCP stream to identify common FTP commands such as `USER`, `PASS`, `LIST`, `RETR`, and `STOR`. Each of these commands corresponds to specific actions, such as logging in (`USER` and `PASS`), listing files on the server (`LIST`), downloading a file (`RETR`), or uploading a file (`STOR`).
   - **Why This Is Important**: Understanding these commands is critical in cybersecurity because it helps me recognize what operations are being performed over the network. For example, spotting an unauthorized `RETR` command might indicate data exfiltration.

## Summary

In this project, I used Wireshark to capture and analyze FTP traffic on a Windows machine. Through the analysis, I observed that FTP transmits data, including usernames and passwords, in plain text, making it susceptible to interception and attacks. The lack of encryption in FTP traffic is a significant security risk, as it allows sensitive information to be easily read by anyone with network access. This project underscores the importance of using secure file transfer protocols like SFTP or FTPS, which provide encryption and better protect data during transmission.