Proj 3x: Harvesting Files from Packet Captures with Wireshark (10 pts.)

What you need:

- A computer (any OS, real or virtual) with an Internet connection

Purpose

You will use Wireshark to collect files from a packet capture.

Stop your Antivirus

This is a real Java attack I performed with Metasploit. It's not very dangerous, because it has a hard-coded attacker IP address of 192.168.198.135 in it.

Unless you have a real attacker at that IP address, running this file won't do any harm. But it WILL set off antivirus software, which will prevent you completing the project.

So disable your antivirus, or use a virtual machine without any antivirus installed.

Downloading the Packet Capture to Examine

Download this file and save it on your desktop:

- pX12-121.pcap (1.2 MB)

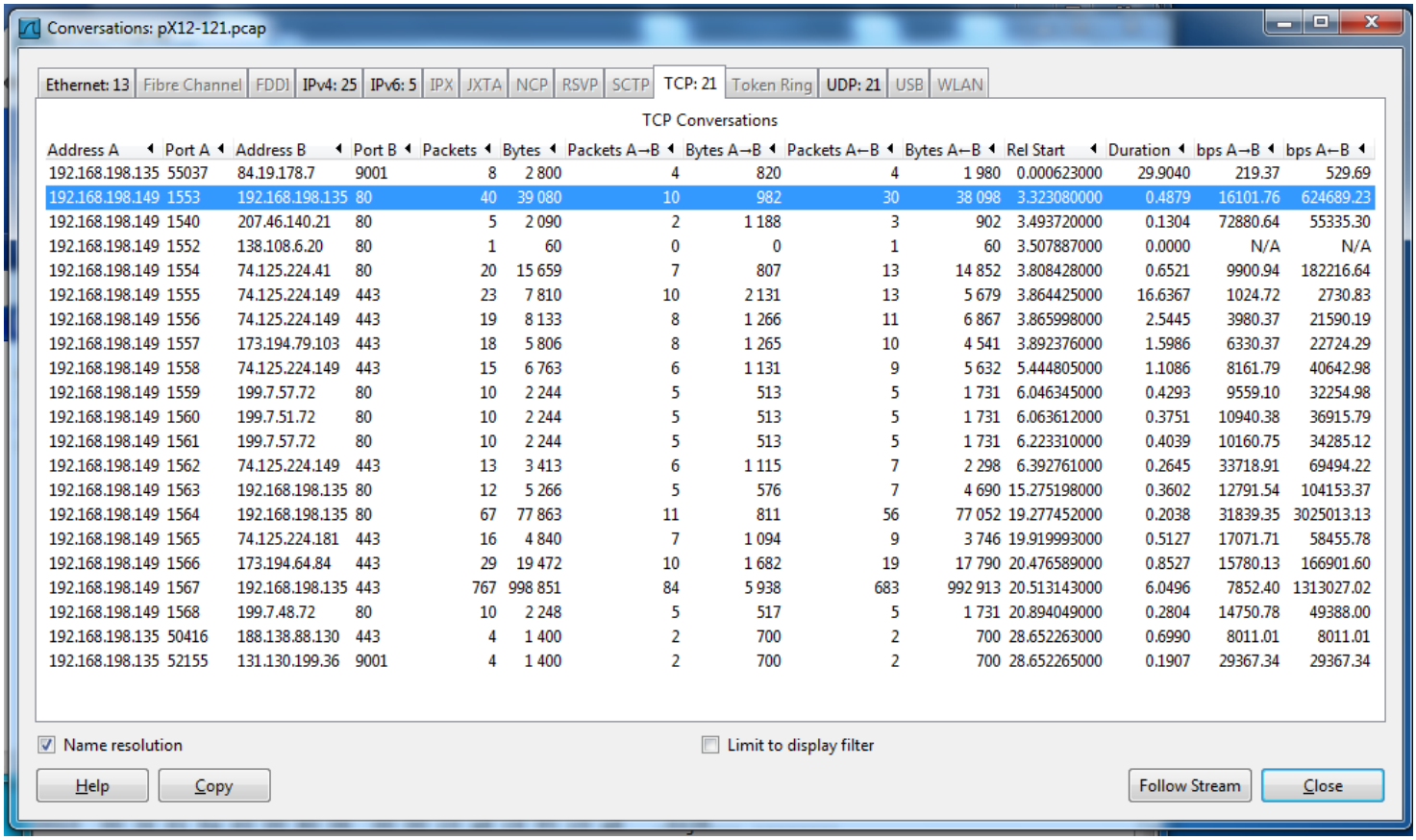
Installing Wireshark

If you don't have Wireshark, open a Web browser and go to http://www.wireshark.org/ to get the appropriate version for your system. Download and install it.

Loading the Packet Capture in Wireshark

Start Wireshark. From the Wireshark menu bar, click **File**, **Open**. Navigate to your desktop and double-click the **pX12-121.pcap** file.

From the Wireshark menu bar, click **Statistics**, **Conversations**. In the "Conversations: pX12-121.pcap" window, click the **TCP:21** tab. You see the 21 conversations in the capture, as shown below:

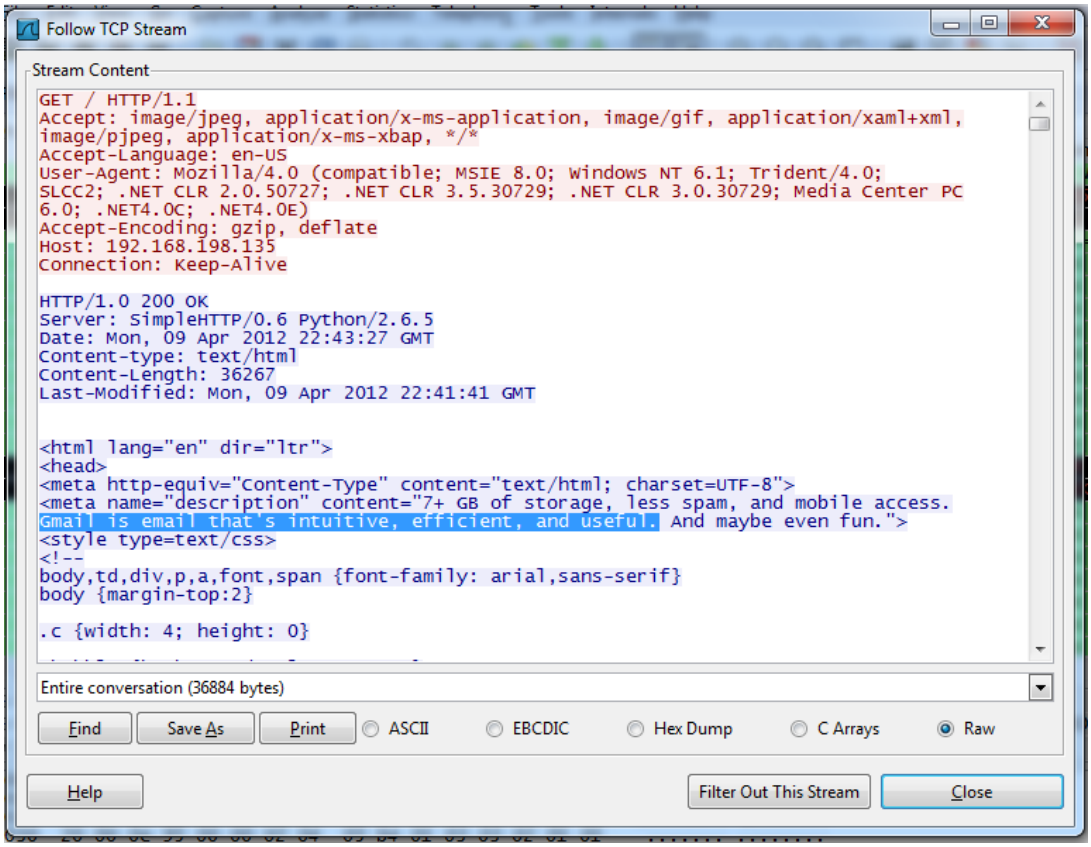


Address A	Port A	Address B	Port B	Packets	Bytes	Packets A→B	Bytes A→B	Packets A←B	Bytes A←B	Rel Start	Duration	bps A→B	bps A←B
192.168.198.135	55037	84.19.178.7	9001	8	2 800	4	820	4	1 980	0.000623000	29.9040	219.37	529.69
192.168.198.149	1553	192.168.198.135	80	40	39 080	10	982	30	38 098	3.323080000	0.4879	16101.76	624689.23
192.168.198.149	1540	207.46.140.21	80	5	2 090	2	1 188	3	902	3.493720000	0.1304	72880.64	55335.30
192.168.198.149	1552	138.108.6.20	80	1	60	0	0	1	60	3.507887000	0.0000	N/A	N/A
192.168.198.149	1554	74.125.224.41	80	20	15 659	7	807	13	14 852	3.808428000	0.6521	9900.94	182216.64
192.168.198.149	1555	74.125.224.149	443	23	7 810	10	2 131	13	5 679	3.864425000	16.6367	1024.72	2730.83
192.168.198.149	1556	74.125.224.149	443	19	8 133	8	1 266	11	6 867	3.865998000	2.5445	3980.37	21590.19
192.168.198.149	1557	173.194.79.103	443	18	5 806	8	1 265	10	4 541	3.892376000	1.5986	6330.37	22724.29
192.168.198.149	1558	74.125.224.149	443	15	6 763	6	1 131	9	5 632	5.444805000	1.1086	8161.79	40642.98
192.168.198.149	1559	199.7.57.72	80	10	2 244	5	513	5	1 731	6.046345000	0.4293	9559.10	32254.98
192.168.198.149	1560	199.7.51.72	80	10	2 244	5	513	5	1 731	6.063612000	0.3751	10940.38	36915.79
192.168.198.149	1561	199.7.57.72	80	10	2 244	5	513	5	1 731	6.223310000	0.4039	10160.75	34285.12
192.168.198.149	1562	74.125.224.149	443	13	3 413	6	1 115	7	2 298	6.392761000	0.2645	33718.91	69494.22
192.168.198.149	1563	192.168.198.135	80	12	5 266	5	576	7	4 690	15.275198000	0.3602	12791.54	104153.37
192.168.198.149	1564	192.168.198.135	80	67	77 863	11	811	56	77 052	19.277452000	0.2038	31839.35	3025013.13
192.168.198.149	1565	74.125.224.181	443	16	4 840	7	1 094	9	3 746	19.919993000	0.5127	17071.71	58455.78
192.168.198.149	1566	173.194.64.84	443	29	19 472	10	1 682	19	17 790	20.476589000	0.8527	15780.13	166901.60
192.168.198.149	1567	192.168.198.135	443	767	998 851	84	5 938	683	992 913	20.513143000	6.0496	7852.40	1313027.02
192.168.198.149	1568	199.7.48.72	80	10	2 248	5	517	5	1 731	20.894049000	0.2804	14750.78	49388.00
192.168.198.135	50416	188.138.88.130	443	4	1 400	2	700	2	700	28.652263000	0.6990	8011.01	8011.01
192.168.198.135	52155	131.130.199.36	9001	4	1 400	2	700	2	700	28.652265000	0.1907	29367.34	29367.34

Click the second conversation, the one that exchanges 40 packets with 192.168.198.135 on port 80, as shown above. Click the **Follow Stream** button.

A "Follow TCP Stream" box pops up, as shown below. You can see the outgoing data in red--an HTTP GET request. The reply is in blue. This is a Gmail login page--you can see the description of the page highlighted in the image below.

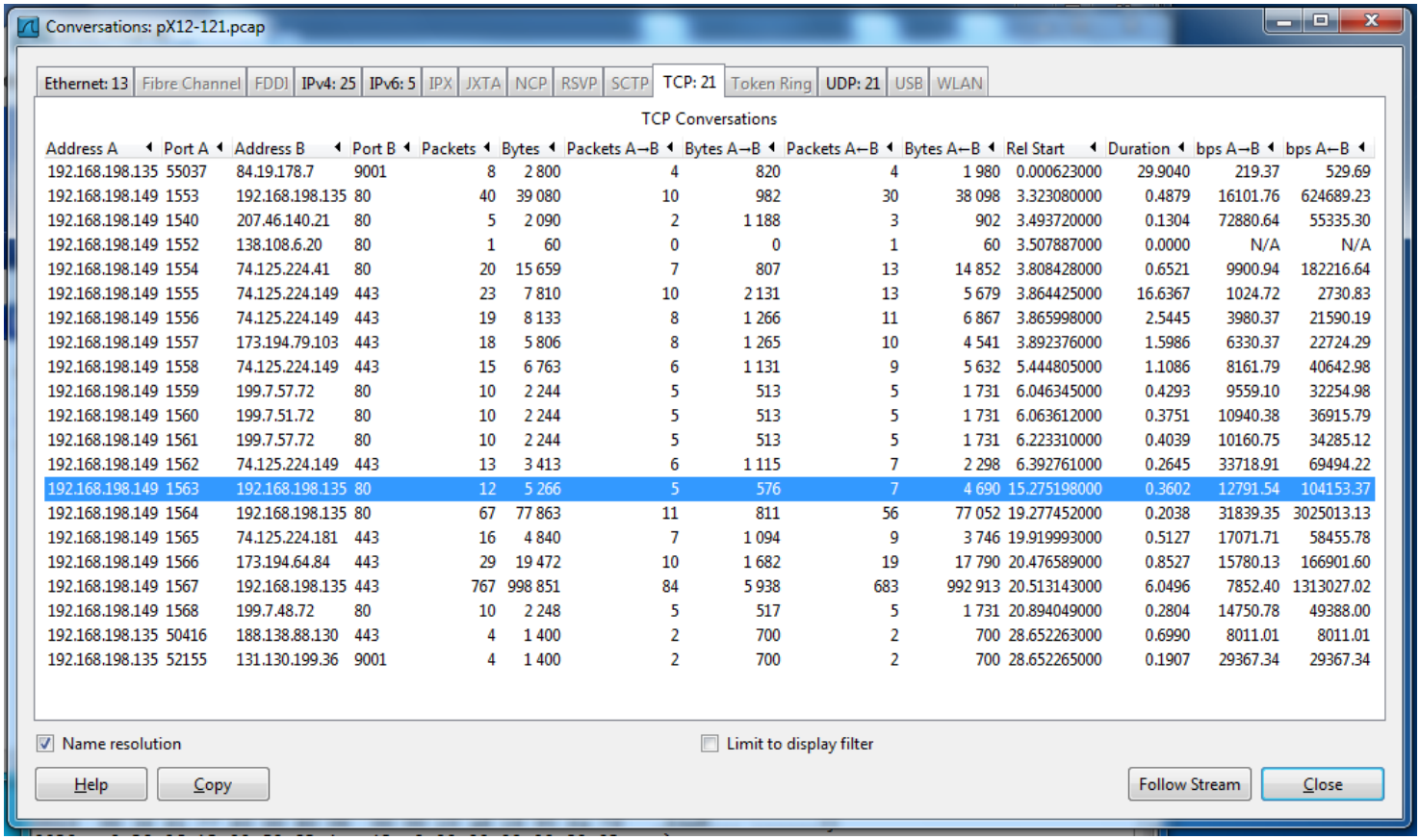
You can already see that this looks suspicious--why is a Gmail login page coming from a private address like that, and not from Google?



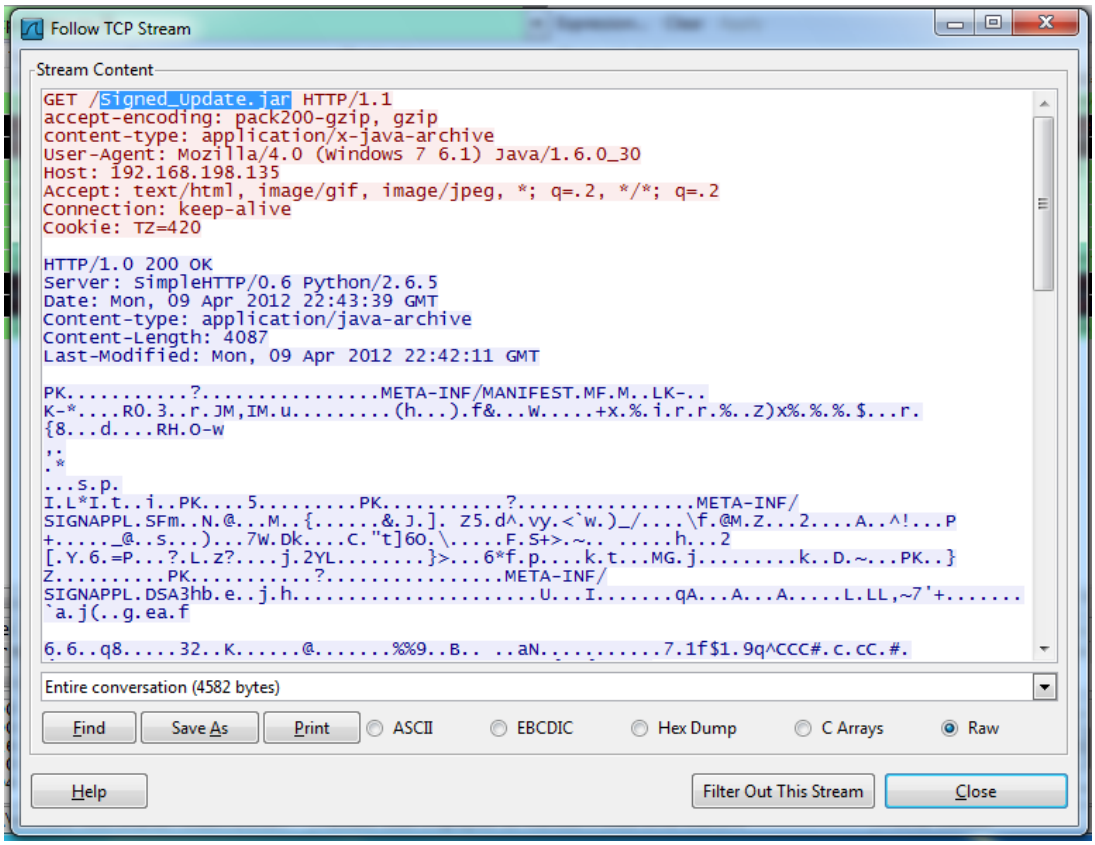
In the "Follow TCP Stream" box, click the **Close** button.

Click the "Conversations: pX12-121.pcap" window to bring it to the front.

Click the conversation that exchanges 12 packets with 192.168.198.135 on port 80. It's near the end of the list, as shown below. Click the **Follow Stream** button.



The "Follow TCP Stream" box shows a request for a file named "Signed_update.jar", as highlighted in the image below.



This is a Java archive. And the real Gmail page doesn't use Java. Also, if it is really a Java Update, it should be coming from Oracle, not from the same server that delivered the Gmail login page.

You can see the file contents in blue in the image above--it's a binary file so it's unreadable.

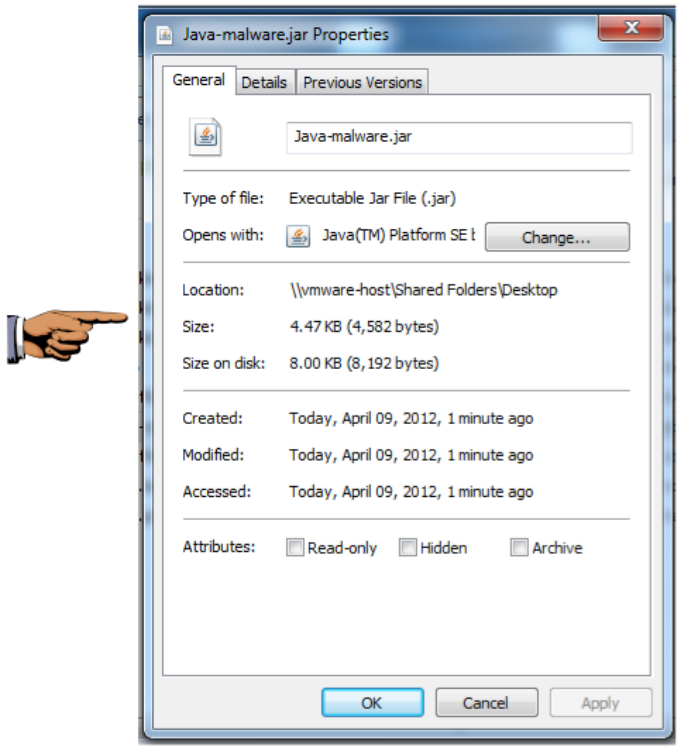
You can, however, save the file in its binary form.

In the "Follow TCP Stream" box, click the **Save As** button. Save the file on your desktop as "**Java-malware.jar**"

Checking the Size of the Java Malware File

On your desktop, right-click the "**Java-malware.jar**" file and click **Properties**.

The file size should be exactly **4,582 bytes**, as shown below:



Saving the Image

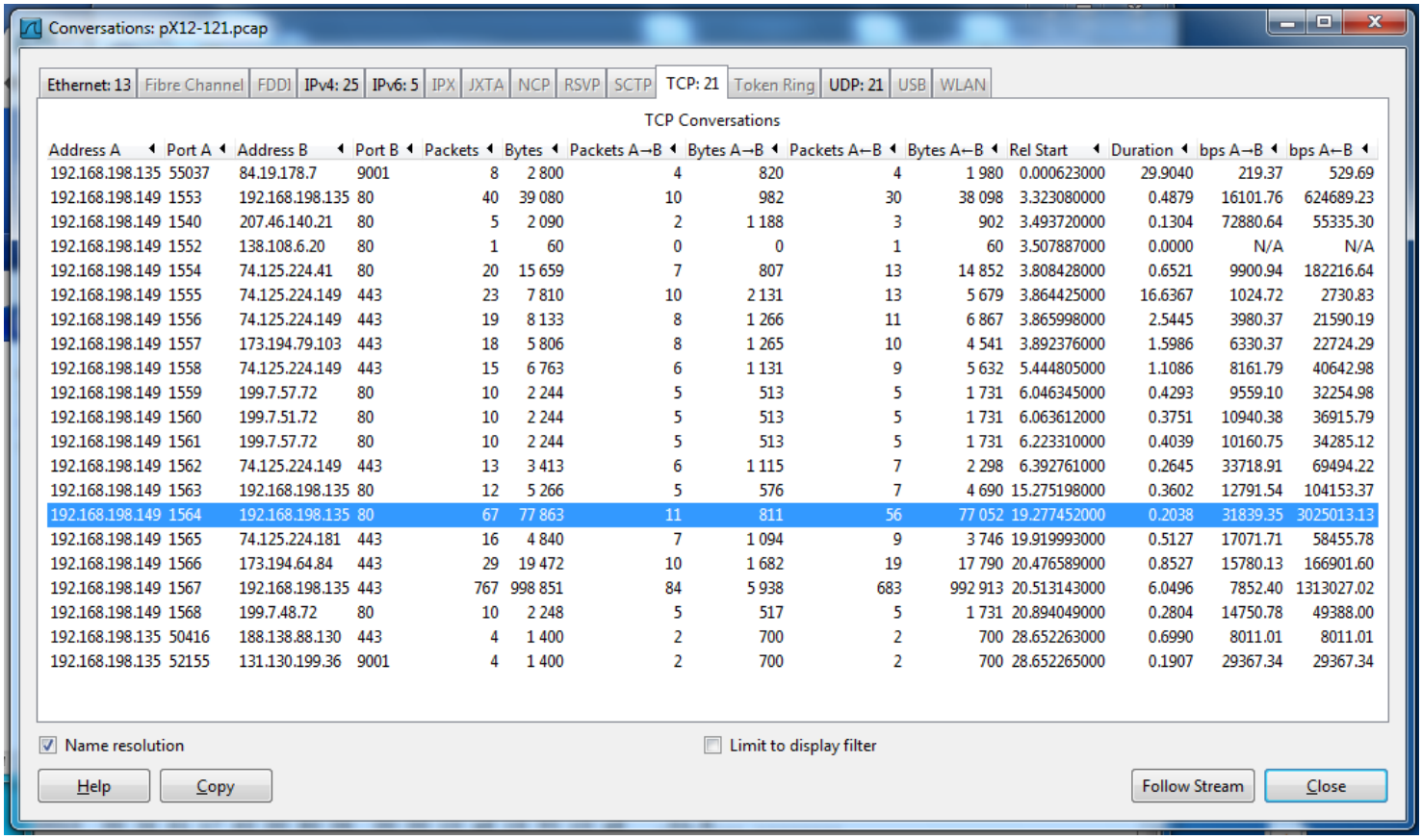
Make sure the file size of **4,582 bytes** is visible.

Save this image with the filename **Proj 3xa from YOUR NAME**

In the "Follow TCP Stream" box, click the **Close** button.

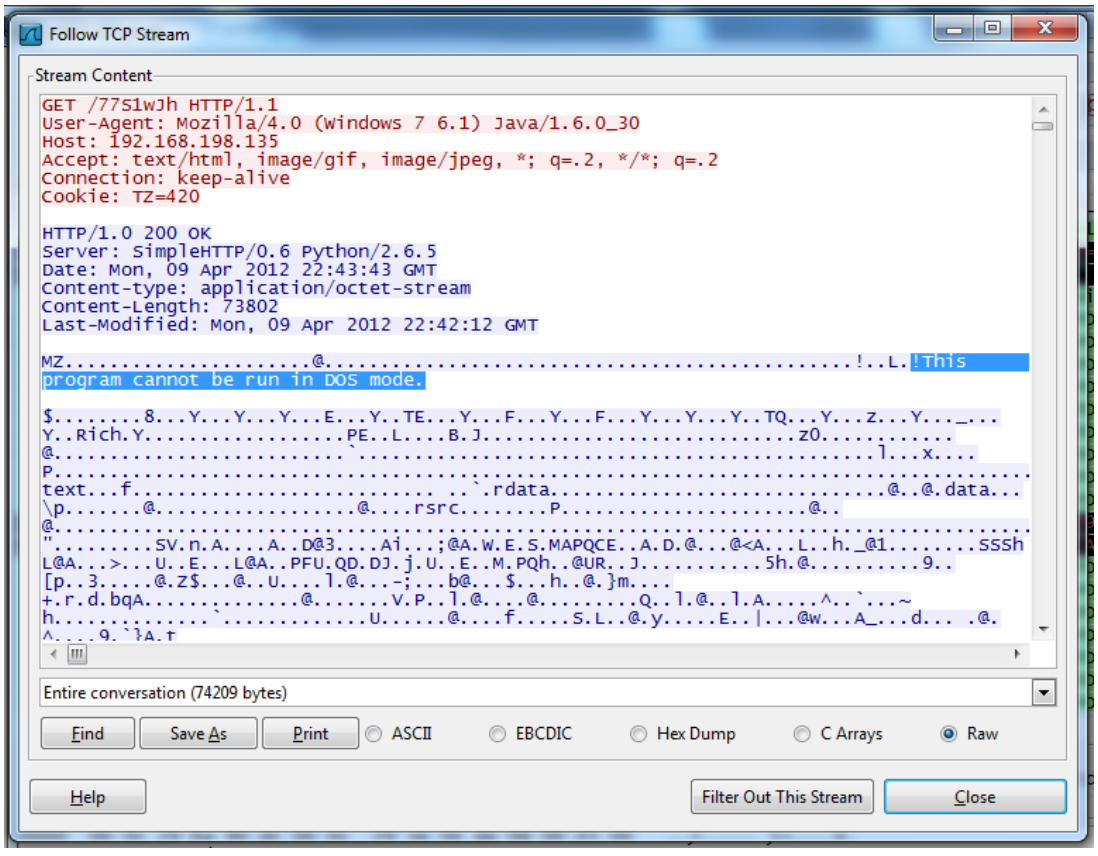
Click the "Conversations: pX12-121.pcap" window to bring it to the front.

Click the conversation that exchanges 67 packets with 192.168.198.135 on port 80. It's near the end of the list, as shown below.



Click the **Follow Stream** button.

This conversation requests a file named "77S1wJh" -- it's not obvious what it is. But the blue response shows a clue--the text "This program cannot be run on DOS mode" is readable, as highlighted in the image below.



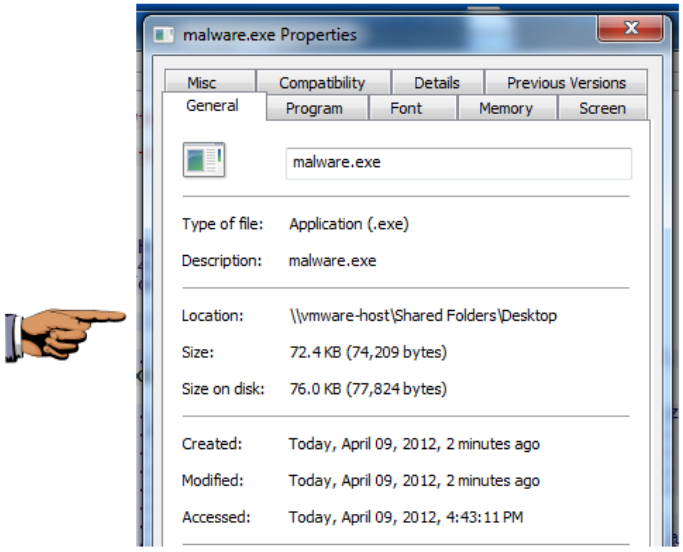
This is a Windows executable file.

In the "Follow TCP Stream" box, click the **Save As** button. Save the file on your desktop as "**malware.exe**"

Checking the Size of the Executable Malware File

On your desktop, right-click the "**malware.exe**" file and click **Properties**.

The file size should be exactly **74,209 bytes**, as shown below:



Saving the Image

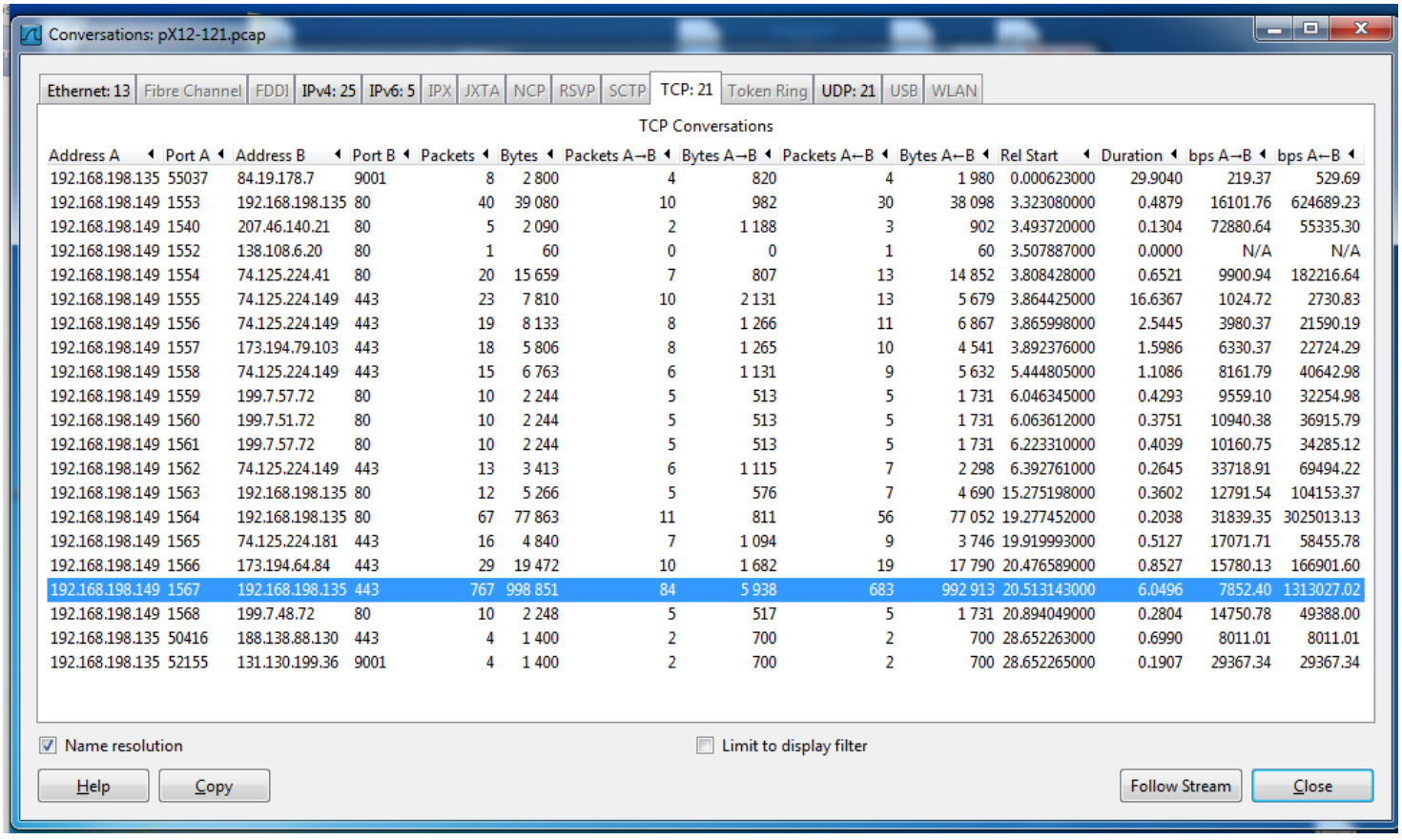
Make sure the file size of **74,209 bytes** is visible.

Save this image with the filename **Proj 3xb from YOUR NAME**

In the "Follow TCP Stream" box, click the **Close** button.

Click the "Conversations: pX12-121.pcap" window to bring it to the front.

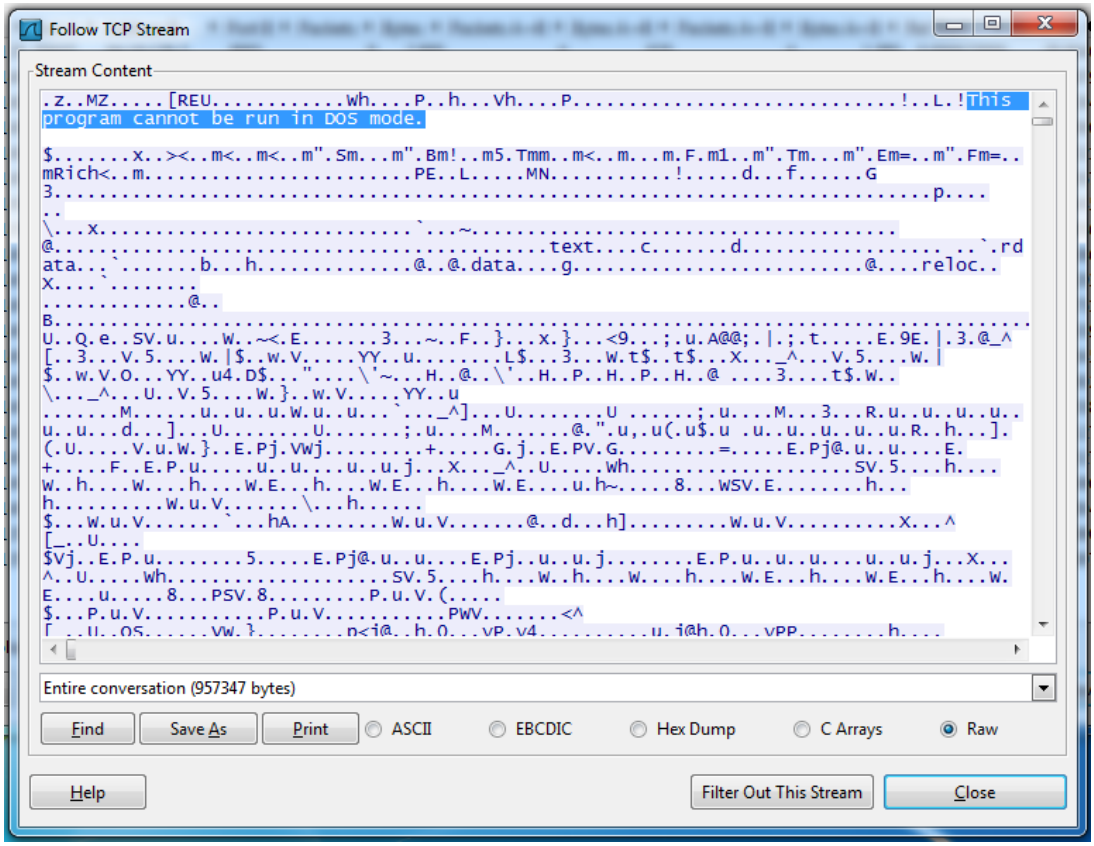
Click the conversation that exchanges 767 packets with 192.168.198.135 on port 443. It's near the end of the list, as shown below.



Click the **Follow Stream** button.

The target machine was under hostile control by this time, so it was sent a file without needing to send an HTTP request first.

And, as you can see below, even though this traffic is coming in from port 443, it is not encrypted--you can read the usual message that indicates a Windows executable file, as shown below:

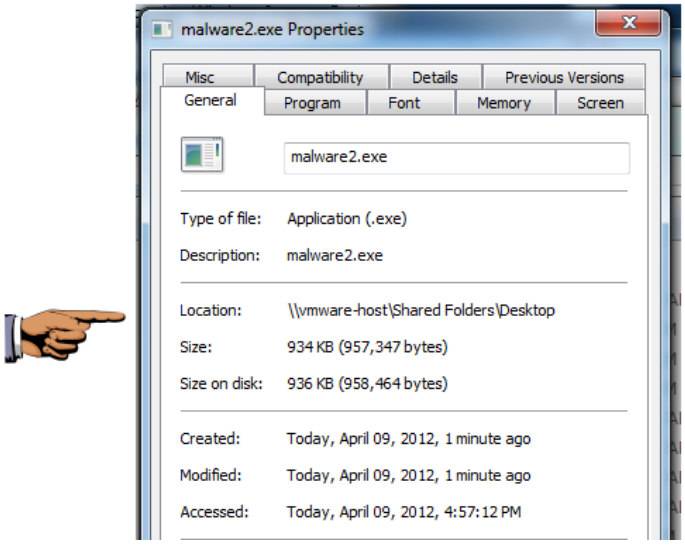


In the "Follow TCP Stream" box, click the **Save As** button. Save the file on your desktop as "**malware2.exe**"

Checking the Size of the Executable Malware File

On your desktop, right-click the "**malware2.exe**" file and click **Properties**.

The file size should be exactly **957,347 bytes**, as shown below:



Saving the Image

Make sure the file size of **957,347 bytes** is visible.

Save this image with the filename **Proj 3xc from YOUR NAME**

Turning in your Project

Email the images to cnit.126sam@gmail.com with the subject line: **Proj 3x from YOUR NAME**

Credits

This is based on a class I took at the HoneyNet conference, from Felix Leder.

Last modified 8-19-13