Wireshark Lab 1 - Intro

IT 520-A – Computer Security Assignment 1
Due Date: January 30th, 2018 (Handed in at the beginning of class)
Instruction: Scroll to the bottom of the page, for in-class deliverables.

In this first lab, you'll get acquainted with Wireshark, and make some simple packet captures and observations.

We will be using the Wireshark packet sniffer [http://www.wireshark.org/] for this and subsequent labs, allowing us to display the contents of messages being sent/received from/by protocols at different levels of the protocol stack. (Technically speaking, Wireshark is a packet analyzer that uses a packet capture library in your computer). Wireshark is a free network protocol analyzer that runs on Windows, Mac, and Linux/Unix computer. It's an ideal packet analyzer for our labs – it is stable, has a large user base and well-documented support that includes a user-guide (http://www.wireshark.org/docs/wsug_html_chunked/), man pages (http://www.wireshark.org/docs/man-pages/), and a detailed FAQ (http://www.wireshark.org/faq.html), rich functionality that includes the capability to analyze hundreds of protocols, and a well-designed user interface. It operates in computers using Ethernet, serial (PPP and SLIP), 802.11 wireless LANs, and many other link-layer technologies (if the OS on which it's running allows Wireshark to do so).

Getting and Installing Wireshark

In order to run Wireshark, you will need to have access to a computer that supports both Wireshark and the *libpcap* or *WinPCap* packet capture library. The *libpcap* software will be installed for you, if it is not installed within your operating system, when you install Wireshark. See http://www.wireshark.org/download.html for a list of supported operating systems and download sites

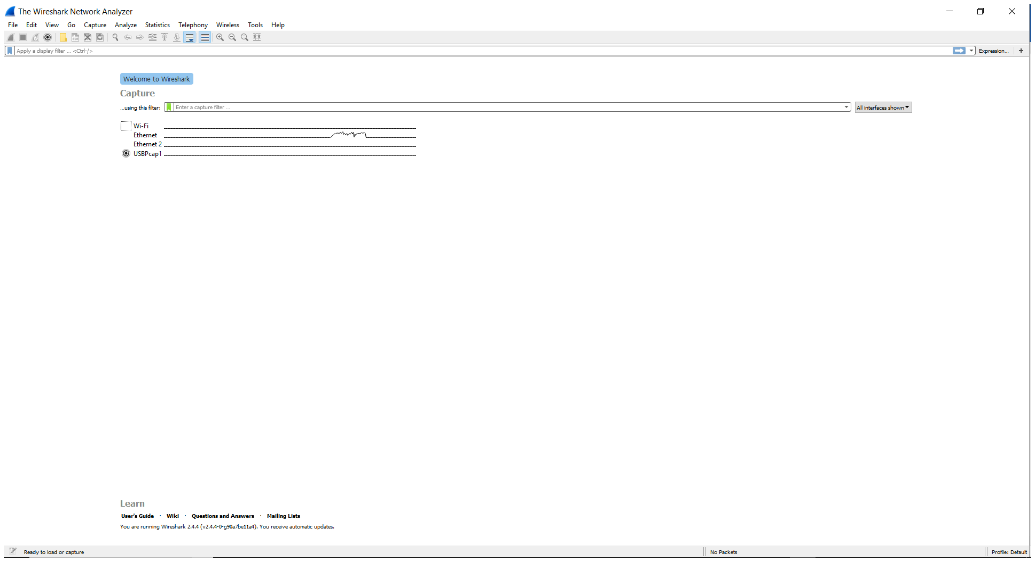
Download and install the Wireshark software:

Go to http://www.wireshark.org/download.html and download and install the Wireshark binary for your computer. The Wireshark FAQ has a number of helpful hints and interesting tidbits of information, particularly if you have trouble installing or running Wireshark.

Hint: There are a lot of videos on YouTube showing how to install Wireshark.

Running Wireshark

When you run the Wireshark program, you'll get a startup screen that looks something like the Figure 1 below. Different versions of Wireshark will have different startup screens – so don't panic if yours doesn't look exactly like the screen below! Wireshark runs on many different platforms with many different window managers, different styles applied and there are different versions of the underlying GUI toolkit used, your screen might look different from the provided screenshots. But as there are no real differences in functionality, these screenshots should still be well understandable.


The Wireshark Network Analyzer
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>
Expression…
+
Welcome to Wireshark
Capture
…using this filter:
Enter a capture filter …
All interfaces shown
Learn
User's Guide · Wiki · Questions and Answers · Mailing Lists

□ Wi-Fi
Ethernet
Ethernet 2
● USBPcap1

You are running Wireshark 2.4.4 (v2.4.4-0-g90a7be11a4). You receive automatic updates.

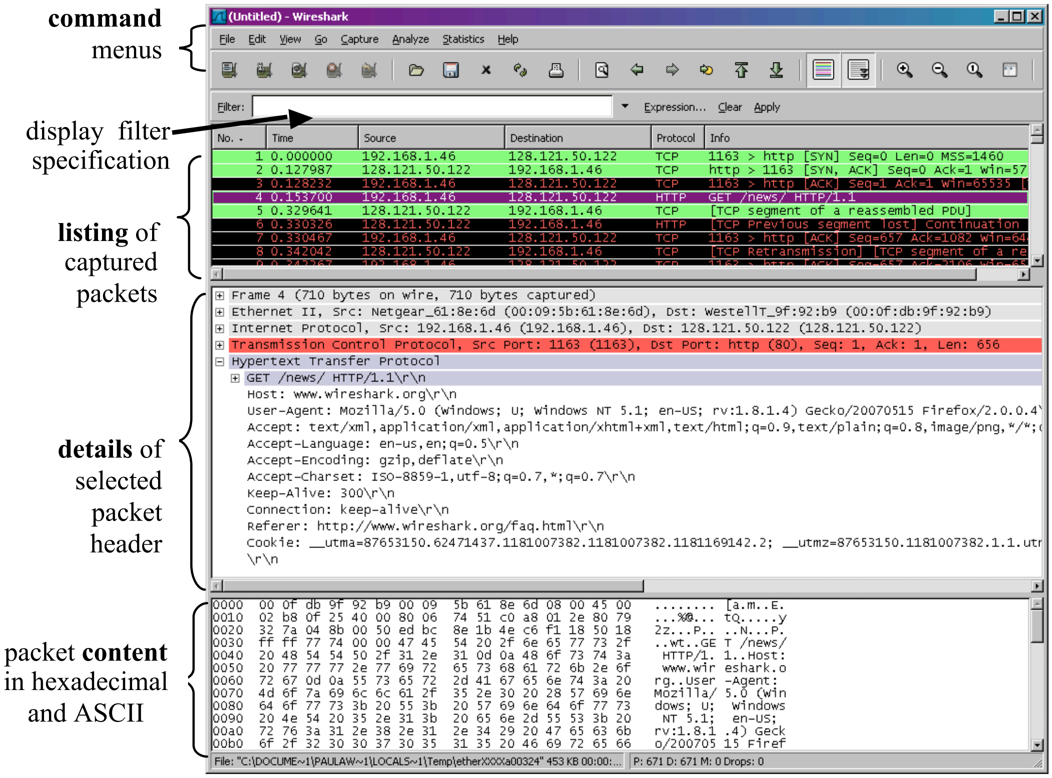
Ready to load or capture | No Packets | Profile: Default

Figure 1: Initial Wireshark Screen

There's not much interesting on this screen. But note that under the Capture section, there is a list of so-called interfaces. The computer we're taking these screenshots from has three real interfaces – "Wi-Fi, Ethernet and Ethernet 2". Yours might look different. Wi-Fi is the interface for Wi-Fi access. Ethernet and Ethernet 2 are the interface for Ethernet ports. All packets to/from this computer will pass through either the Wi-Fi or Ethernet interface, so it's here where we want to capture packets. Double click the interface through which you are getting Internet connectivity.

If you are getting internet access via wireless then select WiFi. If via cable, select Ethernet.

Let's take Wireshark out for a spin! If you click on your connected interface to start packet capture (i.e., for Wireshark to begin capturing all packets being sent to/from that interface), a screen like Figure 2 will be displayed, showing information about the packets being captured. Once you start packet capture, you can stop it by using the stop option on the menu.


command menus
display filter specification
listing of captured packets
details of selected packet header
packet **content** in hexadecimal and ASCII
(Untitled) - Wireshark
File
Edit
View
Go
Capture
Analyze
Statistics
Help
Filter:
Expression...
Clear
Apply

```
No. -  Time       Source            Destination       Protocol  Info
     1 0.000000   192.168.1.46      128.121.50.122    TCP       1163 > http [SYN] Seq=0 Len=0 MSS=1460
     2 0.127987   128.121.50.122    192.168.1.46      TCP       http > 1163 [SYN, ACK] Seq=0 Ack=1 Win=57
     3 0.128232   192.168.1.46      128.121.50.122    TCP       1163 > http [ACK] Seq=1 Ack=1 Win=65535 [
     4 0.153700   192.168.1.46      128.121.50.122    HTTP      GET /news/ HTTP/1.1
     5 0.329641   128.121.50.122    192.168.1.46      TCP       [TCP segment of a reassembled PDU]
     6 0.330326   128.121.50.122    192.168.1.46      HTTP      [TCP Previous segment lost] Continuation
     7 0.330467   192.168.1.46      128.121.50.122    TCP       1163 > http [ACK] Seq=657 Ack=1082 Win=64
     8 0.342042   128.121.50.122    192.168.1.46      TCP       [TCP Retransmission] [TCP segment of a re
     9 0.342267   192.168.1.46      128.121.50.122    TCP       1163 > http [ACK] Seq=657 Ack=2106 Win=65
```

```
⊞ Frame 4 (710 bytes on wire, 710 bytes captured)
⊞ Ethernet II, Src: Netgear_61:8e:6d (00:09:5b:61:8e:6d), Dst: WestellT_9f:92:b9 (00:0f:db:9f:92:b9)
⊞ Internet Protocol, Src: 192.168.1.46 (192.168.1.46), Dst: 128.121.50.122 (128.121.50.122)
⊞ Transmission Control Protocol, Src Port: 1163 (1163), Dst Port: http (80), Seq: 1, Ack: 1, Len: 656
⊟ Hypertext Transfer Protocol
  ⊞ GET /news/ HTTP/1.1\r\n
    Host: www.wireshark.org\r\n
    User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.8.1.4) Gecko/20070515 Firefox/2.0.0.4
    Accept: text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,image/png,*/*;
    Accept-Language: en-us,en;q=0.5\r\n
    Accept-Encoding: gzip,deflate\r\n
    Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7\r\n
    Keep-Alive: 300\r\n
    Connection: keep-alive\r\n
    Referer: http://www.wireshark.org/faq.html\r\n
    Cookie: __utma=87653150.62471437.1181007382.1181007382.1181169142.2; __utmz=87653150.1181007382.1.1.utr
    \r\n
```

```
0000  00 0f db 9f 92 b9 00 09  5b 61 8e 6d 08 00 45 00   ........ [a.m..E.
0010  02 b8 0f 25 40 00 80 06  74 51 c0 a8 01 2e 80 79   ...%@... tQ.....y
0020  32 7a 04 8b 00 50 ed bc  8e 1b 4e c6 f1 18 50 18   2z...P.. ..N...P.
0030  ff ff 77 74 00 00 47 45  54 20 2f 6e 65 77 73 2f   ..wt..GE T /news/
0040  20 48 54 54 50 2f 31 2e  31 0d 0a 48 6f 73 74 3a    HTTP/1. 1..Host:
0050  20 77 77 77 2e 77 69 72  65 73 68 61 72 6b 2e 6f    www.wir eshark.o
0060  72 67 0d 0a 55 73 65 72  2d 41 67 65 6e 74 3a 20   rg..User -Agent: 
0070  4d 6f 7a 69 6c 6c 61 2f  35 2e 30 20 28 57 69 6e   Mozilla/ 5.0 (Win
0080  64 6f 77 73 3b 20 55 3b  20 57 69 6e 64 6f 77 73   dows; U;  Windows
0090  20 4e 54 20 35 2e 31 3b  20 65 6e 2d 55 53 3b 20    NT 5.1;  en-US; 
00a0  72 76 3a 31 2e 38 2e 31  2e 34 29 20 47 65 63 6b   rv:1.8.1 .4) Geck
00b0  6f 2f 32 30 30 37 30 35  31 35 20 46 69 72 65 66   o/200705 15 Firef
```

File: "C:\DOCUME~1\PAULAW~1\LOCALS~1\Temp\etherXXXXa00324" 453 KB 00:00:... P: 671 D: 671 M: 0 Drops: 0

Figure 2: Wireshark Graphical User Interface, during packet capture and analysis

This looks more interesting! The Wireshark interface has five major components:

- The **command menus** are standard pulldown menus located at the top of the window. Of interest to us now are the File and Capture menus. The File menu allows you to save captured packet data or open a file containing previously captured packet data and exit the Wireshark application. The Capture menu allows you to begin packet capture.
- The **packet-listing window** displays a one-line summary for each packet captured, including the packet number (assigned by Wireshark; this is *not* a packet number contained in any protocol's header), the time at which the packet was captured, the packet's source and destination addresses, the protocol type, and protocol-specific information contained in the packet. The packet listing can be sorted according to any of these categories by clicking on a column name. The protocol type field lists the highest-level protocol that sent or received this packet, i.e., the protocol that is the source or ultimate sink for this packet.
- The **packet-header details window** provides details about the packet selected (highlighted) in the packet-listing window. (To select a packet in the packet-listing window, place the cursor over the packet's one-line summary in the packet-listing window and click with the left mouse button.). These details include information about the Ethernet frame (assuming the packet was sent/received over an Ethernet interface) and IP datagram that contains this packet. The amount of Ethernet and IP-layer detail displayed can be expanded or minimized by clicking on the plus minus boxes to the left of the Ethernet frame or IP datagram line in the packet details window. If the packet has been carried over TCP or UDP, TCP or UDP details will also be displayed, which can similarly be

expanded or minimized. Finally, details about the highest-level protocol that sent or received this packet are also provided.

- The **packet-contents window** displays the entire contents of the captured frame, in both ASCII and hexadecimal format.
- Towards the top of the Wireshark graphical user interface, is the **packet display filter field,** into which a protocol name or other information can be entered in order to filter the information displayed in the packet-listing window (and hence the packet-header and packet-contents windows). In the example below, we'll use the packet-display filter field to have Wireshark hide (not display) packets except those that correspond to HTTP messages.

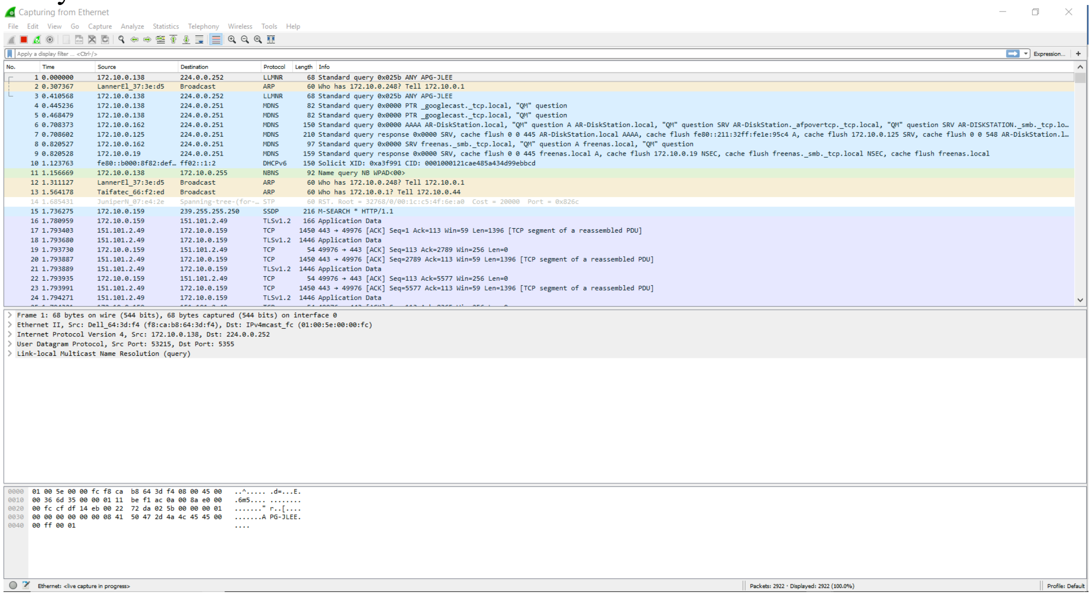
Taking Wireshark for a Test Run

The best way to learn about any new piece of software is to try it out! We'll assume that your computer is connected to the Internet via a wired Ethernet interface. (Don't worry, same steps apply to Wireless connection). Close the Wireshark application, and all your browsers before starting 1 below:

1. Start up your favorite web browser, which will display your selected homepage.

2. Start up the Wireshark software. You will initially see a window like that shown in Figure 1. Wireshark has not yet begun capturing packets.

3. Double click on your interface to start capturing packets. Wireshark is now capturing all packets being sent/received from/by your computer! Your screen should look like figure 3 below. You can always stop by selecting "Stop capturing packets" from the menu. But don't stop packet capture yet. Let's capture some interesting packets first. To do so, we'll need to generate some network traffic. Let's do so using a web browser, which will use the HTTP protocol that we will study in detail in class to download content from a website.


Capturing from Ethernet
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
Expression...
+
Profile: Default

```
No.  Time      Source                 Destination            Protocol Length Info
   1 0.000000  172.10.0.138           224.0.0.252            LLMNR      68 Standard query 0x025b ANY APG-JLEE
   2 0.307367  LannerEl_37:3e:d5      Broadcast              ARP        60 Who has 172.10.0.248? Tell 172.10.0.1
   3 0.410568  172.10.0.138           224.0.0.252            LLMNR      68 Standard query 0x025b ANY APG-JLEE
   4 0.445236  172.10.0.138           224.0.0.251            MDNS       82 Standard query 0x0000 PTR _googlecast._tcp.local, "QM" question
   5 0.468479  172.10.0.138           224.0.0.251            MDNS       82 Standard query 0x0000 PTR _googlecast._tcp.local, "QM" question
   6 0.708373  172.10.0.162           224.0.0.251            MDNS      150 Standard query 0x0000 AAAA AR-DiskStation.local, "QM" question A AR-DiskStation.local, "QM" question SRV AR-DiskStation._afpovertcp._tcp.local, "QM" question SRV AR-DISKSTATION._smb._tcp.lo…
   7 0.708602  172.10.0.125           224.0.0.251            MDNS      210 Standard query response 0x0000 SRV, cache flush 0 0 445 AR-DiskStation.local AAAA, cache flush fe80::211:32ff:fe1e:95c4 A, cache flush 172.10.0.125 SRV, cache flush 0 0 548 AR-DiskStation.l…
   8 0.820527  172.10.0.162           224.0.0.251            MDNS       97 Standard query 0x0000 SRV freenas._smb._tcp.local, "QM" question A freenas.local, "QM" question
   9 0.820528  172.10.0.19            224.0.0.251            MDNS      159 Standard query response 0x0000 SRV, cache flush 0 0 445 freenas.local A, cache flush 172.10.0.19 NSEC, cache flush freenas._smb._tcp.local NSEC, cache flush freenas.local
  10 1.123763  fe80::b000:8f82:def…   ff02::1:2              DHCPv6    150 Solicit XID: 0xa3f991 CID: 0001000121cae485a434d99ebbcd
  11 1.156669  172.10.0.138           172.10.0.255           NBNS       92 Name query NB WPAD<00>
  12 1.311127  LannerEl_37:3e:d5      Broadcast              ARP        60 Who has 172.10.0.248? Tell 172.10.0.1
  13 1.564178  Taifatec_66:f2:ed      Broadcast              ARP        60 Who has 172.10.0.1? Tell 172.10.0.44
  14 1.685431  JuniperN_07:e4:2e      Spanning-tree-(for-…   STP        60 RST. Root = 32768/0/00:1c:c5:4f:6e:a0  Cost = 20000  Port = 0x826c
  15 1.736275  172.10.0.159           239.255.255.250        SSDP      216 M-SEARCH * HTTP/1.1
  16 1.780959  172.10.0.159           151.101.2.49           TLSv1.2   166 Application Data
  17 1.793403  151.101.2.49           172.10.0.159           TCP      1450 443 → 49976 [ACK] Seq=1 Ack=113 Win=59 Len=1396 [TCP segment of a reassembled PDU]
  18 1.793680  151.101.2.49           172.10.0.159           TLSv1.2  1446 Application Data
  19 1.793730  172.10.0.159           151.101.2.49           TCP        54 49976 → 443 [ACK] Seq=113 Ack=2789 Win=256 Len=0
  20 1.793887  151.101.2.49           172.10.0.159           TCP      1450 443 → 49976 [ACK] Seq=2789 Ack=113 Win=59 Len=1396 [TCP segment of a reassembled PDU]
  21 1.793889  151.101.2.49           172.10.0.159           TLSv1.2  1446 Application Data
  22 1.793935  172.10.0.159           151.101.2.49           TCP        54 49976 → 443 [ACK] Seq=113 Ack=5577 Win=256 Len=0
  23 1.793991  151.101.2.49           172.10.0.159           TCP      1450 443 → 49976 [ACK] Seq=5577 Ack=113 Win=59 Len=1396 [TCP segment of a reassembled PDU]
  24 1.794271  151.101.2.49           172.10.0.159           TLSv1.2  1446 Application Data
```

```
> Frame 1: 68 bytes on wire (544 bits), 68 bytes captured (544 bits) on interface 0
> Ethernet II, Src: Dell_64:3d:f4 (f8:ca:b8:64:3d:f4), Dst: IPv4mcast_fc (01:00:5e:00:00:fc)
> Internet Protocol Version 4, Src: 172.10.0.138, Dst: 224.0.0.252
> User Datagram Protocol, Src Port: 53215, Dst Port: 5355
> Link-local Multicast Name Resolution (query)
```

```
0000  01 00 5e 00 00 fc f8 ca  b8 64 3d f4 08 00 45 00   ..^..... .d=...E.
0010  00 36 6d 35 00 00 01 11  be f1 ac 0a 00 8a e0 00   .6m5.... ........
0020  00 fc cf df 14 eb 00 22  72 da 02 5b 00 00 00 01   ......." r..[....
0030  00 00 00 00 00 00 08 41  50 47 2d 4a 4c 45 45 00   .......A PG-JLEE.
0040  00 ff 00 01                                        ....
```

```
Ethernet: <live capture in progress>          Packets: 2922 · Displayed: 2922 (100.0%)
```

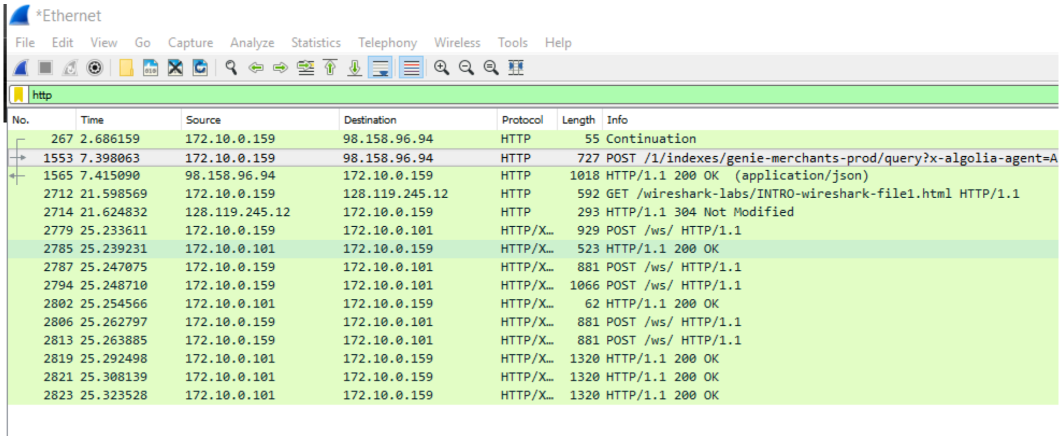
Figure 3: Wireshark Capture

4. While Wireshark is running, enter the URL: http://gaia.cs.umass.edu/wireshark-labs/INTRO-wireshark-file1.html and have that page displayed in your browser.

5. After your browser has displayed the INTRO-wireshark-file1.html page (it is a simple one line of congratulations), stop Wireshark packet capture by selecting stop in the Wireshark capture window. The main Wireshark window should now look similar to Figure 3.

6. You now have live packet data that contains all protocol messages exchanged between your computer and other network entities! The HTTP message exchanges with the gaia.cs.umass.edu web server should appear somewhere in the listing of packets captured. But there will be many other types of packets displayed as well (see, e.g., the many different protocol types shown in the *Protocol* column in Figure 3). Even though the only action you took was to download a web page, there were evidently many other protocols running on your computer that are unseen by the user. We'll learn much more about these protocols as we progress through the text! For now, you should just be aware that there is often much more going on than "meet's the eye"!

7. Type in "http" (without the quotes, and in lower case – all protocol names are in lower case in Wireshark) into the display filter specification window at the top of the main Wireshark window. Then select *Apply* (to the right of where you entered "http"). This will cause only HTTP message to be displayed in the packet-listing window like figure 4 below:



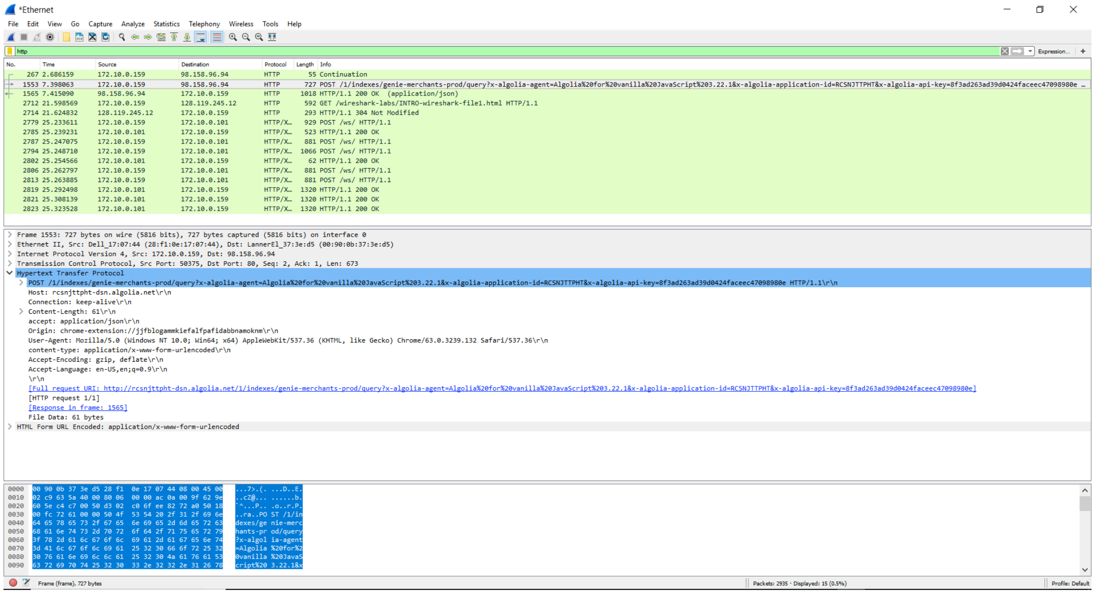
No.	Time	Source	Destination	Protocol	Length	Info
267	2.686159	172.10.0.159	98.158.96.94	HTTP	55	Continuation
1553	7.398063	172.10.0.159	98.158.96.94	HTTP	727	POST /1/indexes/genie-merchants-prod/query?x-algolia-agent=A
1565	7.415090	98.158.96.94	172.10.0.159	HTTP	1018	HTTP/1.1 200 OK (application/json)
2712	21.598569	172.10.0.159	128.119.245.12	HTTP	592	GET /wireshark-labs/INTRO-wireshark-file1.html HTTP/1.1
2714	21.624832	128.119.245.12	172.10.0.159	HTTP	293	HTTP/1.1 304 Not Modified
2779	25.233611	172.10.0.159	172.10.0.101	HTTP/X…	929	POST /ws/ HTTP/1.1
2785	25.239231	172.10.0.101	172.10.0.159	HTTP/X…	523	HTTP/1.1 200 OK
2787	25.247075	172.10.0.159	172.10.0.101	HTTP/X…	881	POST /ws/ HTTP/1.1
2794	25.248710	172.10.0.159	172.10.0.101	HTTP/X…	1066	POST /ws/ HTTP/1.1
2802	25.254566	172.10.0.101	172.10.0.159	HTTP/X…	62	HTTP/1.1 200 OK
2806	25.262797	172.10.0.159	172.10.0.101	HTTP/X…	881	POST /ws/ HTTP/1.1
2813	25.263885	172.10.0.159	172.10.0.101	HTTP/X…	881	POST /ws/ HTTP/1.1
2819	25.292498	172.10.0.101	172.10.0.159	HTTP/X…	1320	HTTP/1.1 200 OK
2821	25.308139	172.10.0.101	172.10.0.159	HTTP/X…	1320	HTTP/1.1 200 OK
2823	25.323528	172.10.0.101	172.10.0.159	HTTP/X…	1320	HTTP/1.1 200 OK

Figure 4: Wireshark HTTP Filter.

8. Find the HTTP GET message that was sent from your computer to the gaia.cs.umass.edu HTTP server. (Look for an HTTP GET message in the "listing of captured packets" portion of the Wireshark window (see Figure 4 – the 4th among the list) that shows "GET" followed by the gaia.cs.umass.edu URL that you entered. When you select the HTTP GET message, the Ethernet frame, IP datagram, TCP segment, and HTTP message header information will be displayed

in the packet-header window[1]. By clicking on '+' and '-' right-pointing and down-pointing arrowheads to the left side of the packet details window, *minimize* the amount of Frame, Ethernet, Internet Protocol, and Transmission Control Protocol information displayed. *Maximize* the amount information displayed about the HTTP protocol. Your Wireshark display should now look roughly as shown in Figure 5.

9. Save the file and Exit Wireshark



```
http
```

```
No.     Time          Source            Destination      Protocol  Length  Info
  267   2.686159      172.10.0.159      98.158.96.94     HTTP        55 Continuation
 1553   7.398063      172.10.0.159      98.158.96.94     HTTP       727 POST /1/indexes/genie-merchants-prod/query?x-algolia-agent=Algolia%20for%20vanilla%20JavaScript%203.22.1&x-algolia-application-id=RCSNJTTPHT&x-algolia-api-key=8f3ad263ad39d0424faceec47098980e …
 1565   7.415090      98.158.96.94      172.10.0.159     HTTP      1018 HTTP/1.1 200 OK  (application/json)
 2712  21.598569      172.10.0.159      128.119.245.12   HTTP       592 GET /wireshark-labs/INTRO-wireshark-file1.html HTTP/1.1
 2714  21.624832      128.119.245.12    172.10.0.159     HTTP       293 HTTP/1.1 304 Not Modified
 2779  25.233611      172.10.0.159      172.10.0.101     HTTP/X…    929 POST /ws/ HTTP/1.1
 2785  25.239231      172.10.0.101      172.10.0.159     HTTP/X…    523 HTTP/1.1 200 OK
 2787  25.247075      172.10.0.159      172.10.0.101     HTTP/X…    881 POST /ws/ HTTP/1.1
 2794  25.248710      172.10.0.159      172.10.0.101     HTTP/X…   1066 POST /ws/ HTTP/1.1
 2802  25.254566      172.10.0.101      172.10.0.159     HTTP/X…     62 HTTP/1.1 200 OK
 2806  25.262797      172.10.0.159      172.10.0.101     HTTP/X…    881 POST /ws/ HTTP/1.1
 2813  25.263885      172.10.0.159      172.10.0.101     HTTP/X…    881 POST /ws/ HTTP/1.1
 2819  25.292498      172.10.0.101      172.10.0.159     HTTP/X…   1320 HTTP/1.1 200 OK
 2821  25.308139      172.10.0.101      172.10.0.159     HTTP/X…   1320 HTTP/1.1 200 OK
 2823  25.323528      172.10.0.101      172.10.0.159     HTTP/X…   1320 HTTP/1.1 200 OK
```

```
> Frame 1553: 727 bytes on wire (5816 bits), 727 bytes captured (5816 bits) on interface 0
> Ethernet II, Src: Dell_17:07:44 (28:f1:0e:17:07:44), Dst: LannerEl_37:3e:d5 (00:90:0b:37:3e:d5)
> Internet Protocol Version 4, Src: 172.10.0.159, Dst: 98.158.96.94
> Transmission Control Protocol, Src Port: 50375, Dst Port: 80, Seq: 2, Ack: 1, Len: 673
v Hypertext Transfer Protocol
  > POST /1/indexes/genie-merchants-prod/query?x-algolia-agent=Algolia%20for%20vanilla%20JavaScript%203.22.1&x-algolia-application-id=RCSNJTTPHT&x-algolia-api-key=8f3ad263ad39d0424faceec47098980e HTTP/1.1\r\n
    Host: rcsnjttpht-dsn.algolia.net\r\n
    Connection: keep-alive\r\n
  > Content-Length: 61\r\n
    accept: application/json\r\n
    Origin: chrome-extension://jjfblogammkiefalfpafidabbnamoknm\r\n
    User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/63.0.3239.132 Safari/537.36\r\n
    content-type: application/x-www-form-urlencoded\r\n
    Accept-Encoding: gzip, deflate\r\n
    Accept-Language: en-US,en;q=0.9\r\n
    \r\n
    [Full request URI: http://rcsnjttpht-dsn.algolia.net/1/indexes/genie-merchants-prod/query?x-algolia-agent=Algolia%20for%20vanilla%20JavaScript%203.22.1&x-algolia-application-id=RCSNJTTPHT&x-algolia-api-key=8f3ad263ad39d0424faceec47098980e]
    [HTTP request 1/1]
    [Response in frame: 1565]
    File Data: 61 bytes
> HTML Form URL Encoded: application/x-www-form-urlencoded
```

```
0000  00 90 0b 37 3e d5 28 f1  0e 17 07 44 08 00 45 00   ...7>.(. ...D..E.
0010  02 c9 63 5a 40 00 80 06  00 00 ac 0a 00 9f 62 9e   ..cZ@... ......b.
0020  60 5e c4 c7 00 50 d3 02  c0 6f ee 82 72 a0 50 18   `^...P.. .o..r.P.
0030  00 fc 72 61 00 00 50 4f  53 54 20 2f 31 2f 69 6e   ..ra..PO ST /1/in
0040  64 65 78 65 73 2f 67 65  6e 69 65 2d 6d 65 72 63   dexes/ge nie-merc
0050  68 61 6e 74 73 2d 70 72  6f 64 2f 71 75 65 72 79   hants-pr od/query
0060  3f 78 2d 61 6c 67 6f 6c  69 61 2d 61 67 65 6e 74   ?x-algol ia-agent
0070  3d 41 6c 67 6f 6c 69 61  25 32 30 66 6f 72 25 32   =Algolia %20for%2
0080  30 76 61 6e 69 6c 6c 61  25 32 30 4a 61 76 61 53   0vanilla %20JavaS
0090  63 72 69 70 74 25 32 30  33 2e 32 32 2e 31 26 78   cript%20 3.22.1&x
```

```
Frame (frame), 727 bytes        Packets: 2935 · Displayed: 15 (0.5%)        Profile: Default
```

Figure 5: Wireshark window after step 8

Class deliverable in the next page!!!

What to hand in

For each of these question, take a screen shot and add attach it to your answer. Also, save your Wireshark lab file. We would use it later in the class.

The goal of this first lab was primarily to introduce you to Wireshark. The following questions will demonstrate that you've been able to get Wireshark up and running, and have explored some of its capabilities. Answer the following questions, based on your Wireshark experimentation:

1. What is the Internet address of your computer?
2. List 3 different protocols that appear in the protocol column in the unfiltered packet-listing window in step 7 above.
3. How long did it take from when the HTTP GET message was sent until the HTTP OK reply was received? (By default, the value of the Time column in the packet-listing window is the amount of time, in seconds, since Wireshark tracing began. To display the Time field in time-of-day format, select the Wireshark *View* pull down menu, then select Time *Display Format*, then select *Time-of-day*.)
4. What is the Internet address of the gaia.cs.umass.edu (also known as www-net.cs.umass.edu)?
5. Print the two HTTP messages (GET and OK) referred to in question 2 above. To do so, select *Print* from the Wireshark *File* command menu, and select the "*Selected Packet Only"* and *"Print as displayed"* radial buttons, and then click OK.

Don't forget to save your Wireshark Lab file.