

Thomas Murphy
Cyb 605-Z3 – Principles of Cybersecurity
Wireshark Lab
Oct 3, 2016

Introduction

This lab introduces packet sniffing and packet analysis; the process of capturing any data passed over the local network and looking for any information that may be useful. Packet sniffers can be hardware appliances or software based (Hannah, 2011). A sniffer works by placing the network interface you want to listen to into promiscuous mode, thus reporting all the packets that it sees (Weadock, 2009).

The lab specifically focuses on the use of Wireshark, a graphical packet sniffer, and packet sniffing for security professionals. Wireshark (and tcpdump) enable administrators to view and examine packets in a granular format in either real-time or to a capture file for examination later. This provides a tool for the administrator to find network errors, measure bandwidth, become aware of intrusion attempts and attacks, map and discover network devices, and become aware of protocols and applications being used on the network.

In this lab I'm using CentOS version 7.2 and A VMWare version of Windows 95. I've run Wireshark over my wireless home network with approximately 20 nodes, including cell phones, two Roku devices, a Brother printer, various laptops, and baby monitors. I've also run it on a corporate network (with permission) that contains high speed market data and multicast networking. Lastly I've also captured data at a public wireless hotspot that required no passwords.

Objectives

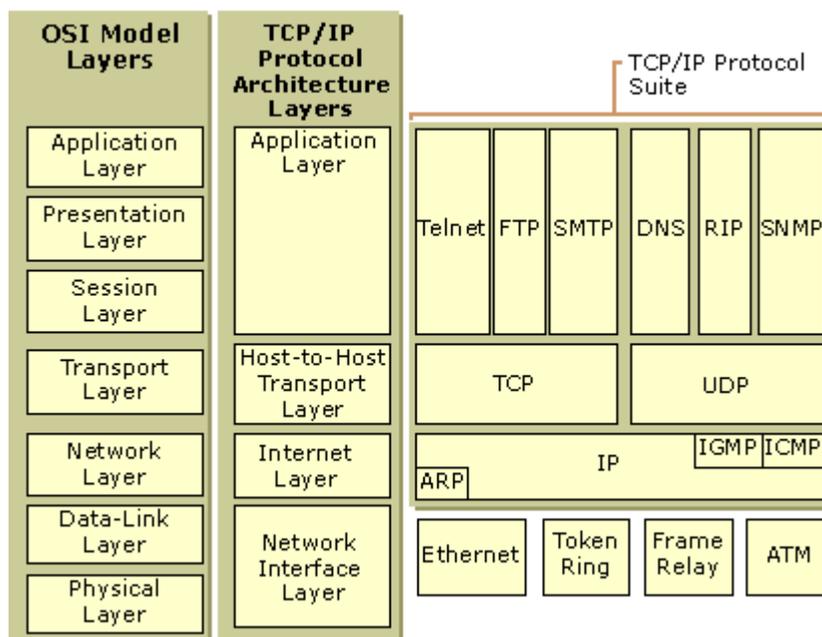
The objectives of this lab are to execute the Wireshark Lab assignments in order to review and enhance previous learning, as well as to learn through experience, and gain the skills and knowledge about various tools currently available. In order to achieve the objective I've connected to three networks using my Dell Laptop running Centos 7.2 through its wireless network interface. I captured packets over a period of time, both for live capture into capture files for later analysis. I applied various filtering to isolate specific types of network packets and described my finding in detail.

Definitions

- Wireshark
 - a) **Sniffer** - software that intercepts and logs network traffic that it can "see" on a network interface. It's used for capturing and reporting data flows. (Paessler, 1996)
 - b) **Protocol** – a set of agreed upon rules for how data is transmitted. In terms of network protocols, there are protocols for how data is packaged and exchanged across a network. The most common of these protocols is "TCP/IP". There are also application protocols that define how a client and server will communicate upon making a network connection (Hunt 1992),
 - c) **Protocol Analyzer** – A tool used to analyze protocol interaction between two systems (Hunt, 1992)
 - d) **Packet** – Data that is communicated across a network is broken into small blocks called packets. "Dividing the data into packets helps the sender and receiver of the data decide

which packets arrived in tact”. “Dividing data into packets also ensures that multiple network connections can share a network device” (Comer, 1997)

- e) **Packet Filter** – A tool used for selectively accepting or blocking data as it passes through a network interface (OpenBSD, 2016)
 - f) **Live Capture** – there are two ways to capture data; live and into a file for analysis later on.
- **TCP/IP**
 - a) **Transport Control Protocol (TCP)** - A connection oriented protocol that provides a full-duplex byte stream for user processes. This means, that when a packet is sent to a host, an acknowledgement packet is returned to the sender. (Stevens, 1990)
 - b) **Internet Protocol (IP)** – This is generally combined with TCP for the purpose of routing and delivering the TCP packet (Stevens, 1990)
 - c) **Network Stack** – A network stack is a reference to the layers of protocols necessary for data communication. When discussing a network stack we generally refer to the OSI model of network communications.

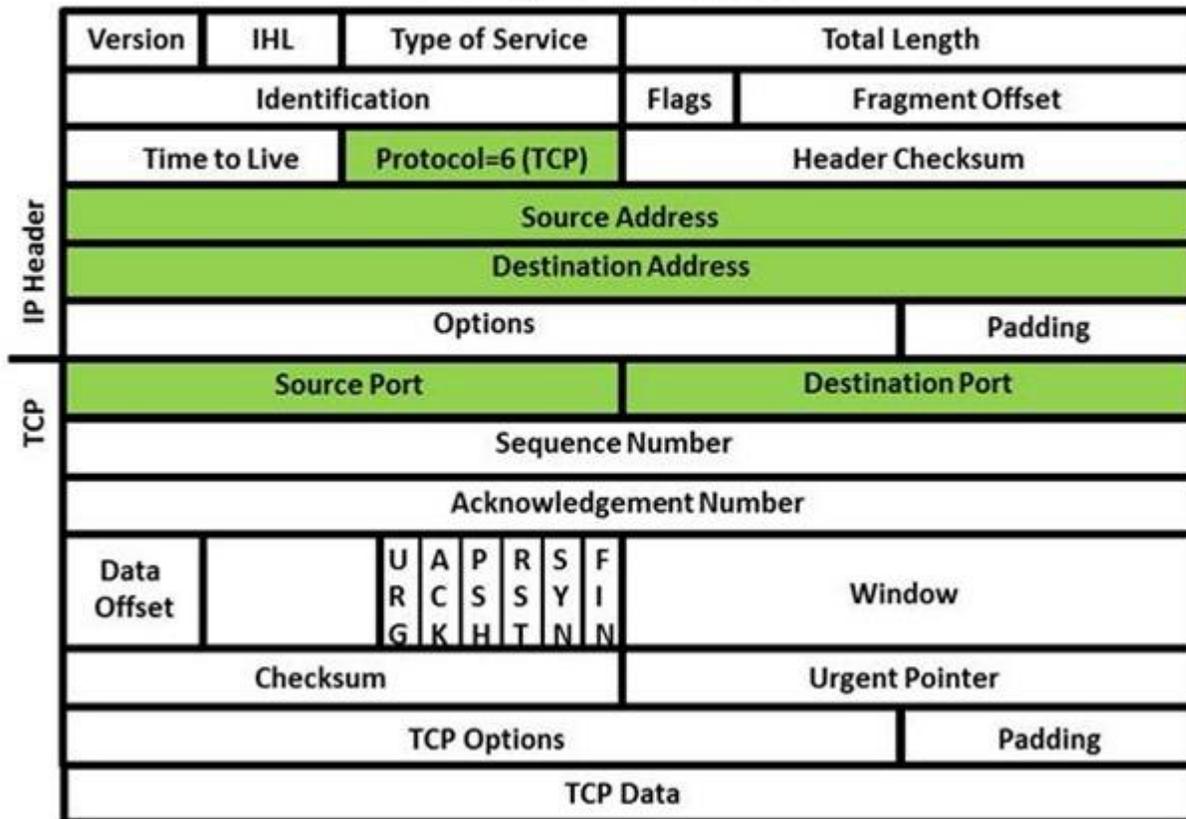


The OSI model is a theoretical model of communication. In the above diagram, you can see how the TCP/IP Network Stack lines up against OSI and shows examples of each layer in the suite. (Erbay, 2016)

- d) **Packet Header** – A packet contains two parts, a header and a payload. The header contains information about where the packet should be sent and details about the contents of the packet, while the payload contains the data that is actually being sent. A typical IPv4 packet contains 20 bytes of data. The 20 bytes of data contain the IP version (4 for IPv4), IP Header Length, type of service, datagram length, an ID tag to

help reconstruct, instructions on fragmentation, fragmentation offset (what part of the packets to start reconstruction), TTL (number of network hops), a header checksum (to help detect errors), source IP address and the destination IP address (Comer, 1997).

TCP/IP Packet



Using Wireshark

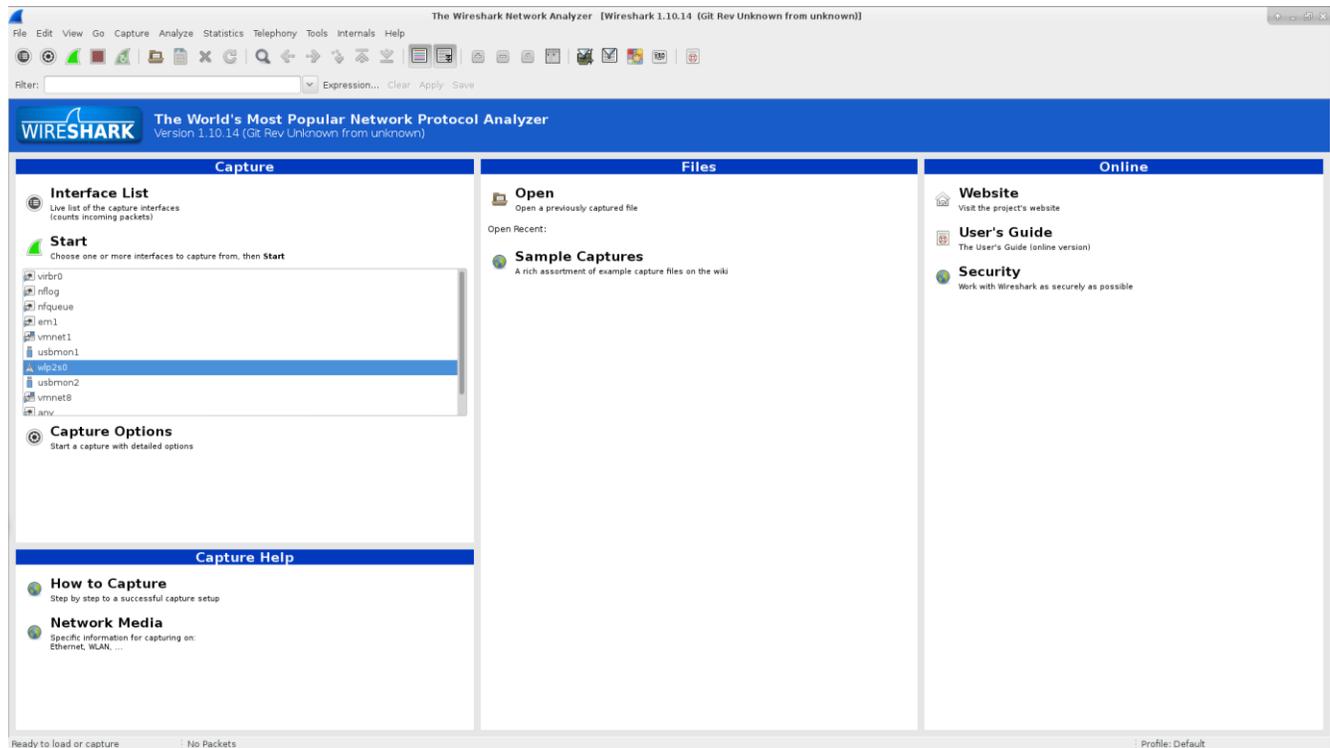
This section was an introduction for starting Wireshark, finding documentation and beginning to sniff data from a network interface.

Procedure:

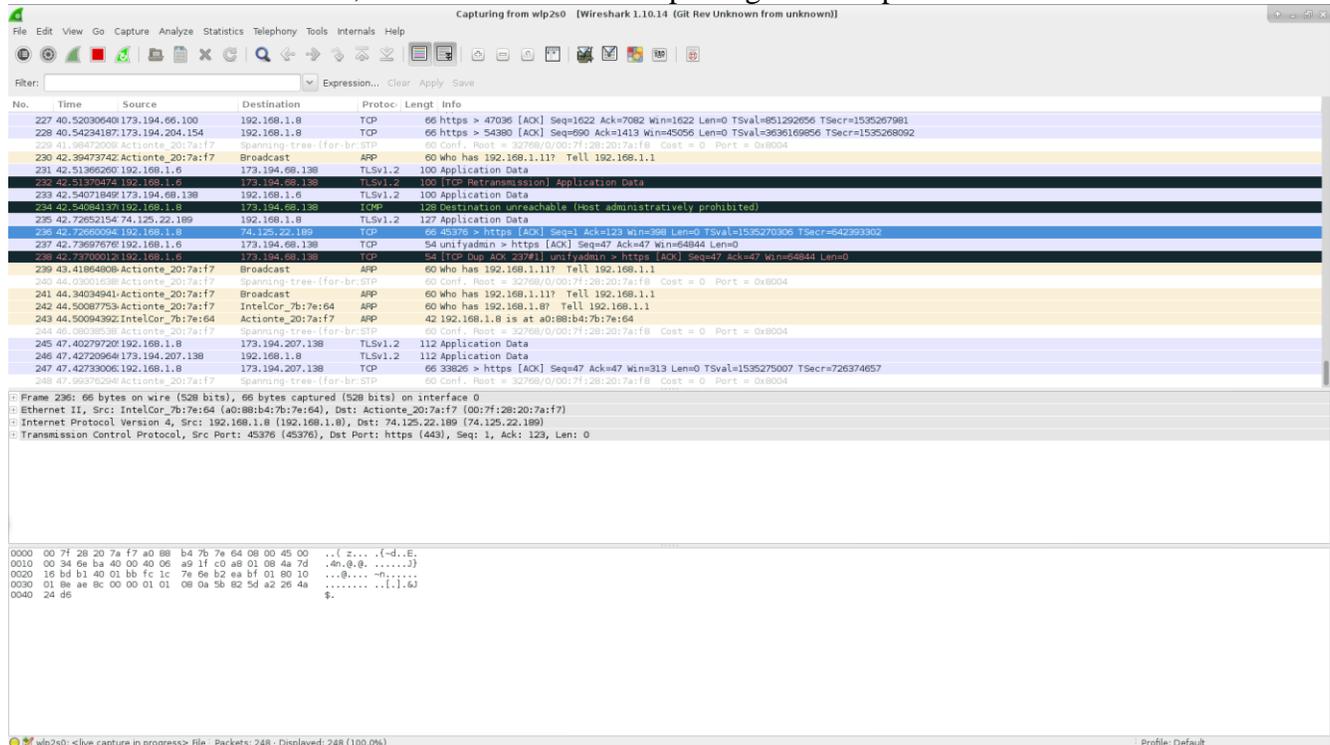
Steps 1 and 2 of the lab are for downloading and installing Wireshark and all other necessary packages. For Wireshark on Centos 7.2 (or any Red Hat Enterprise Linux variant) the command is simply:

```
yum -y install wireshark*
```

Step 3: Upon starting Wireshark we were asked to select our Internet facing interface

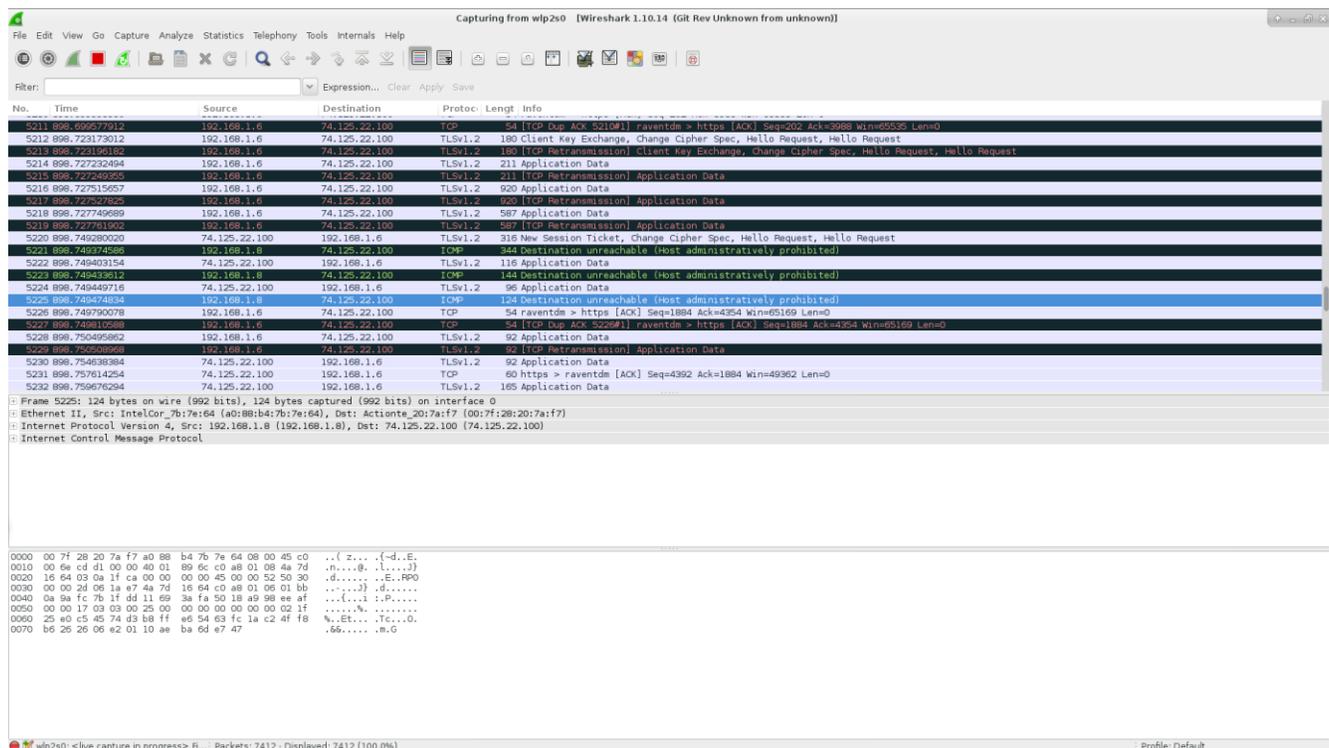


Once the NIC was selected, we were able to start capturing network packets:



From the picture above you can note that the GUI is broken up into three main panels.

- The packet frame** - This displays general contents in the capture file including, the
 - Number off packets.
 - Time - The timestamp of the packet.
 - Source - The address where this packet is coming from.
 - Destination - The address where this packet is going to.
 - Protocol The protocol name in a short version.
 - Length The length of each packet.
 - Additional information about the packet content. (Wireshark, 2014)
- The “Packet Details” pane** - This shows the protocols and the protocol fields of the packet. Of special note, any data enclosed in “[“ and “]” brackets is generated by Wireshark. Additionally, if Wireshark detects a relationship to another packet, it will create a link to that packet (Wireshark, 2014).
- The “Packet Bytes” pane** - This frame contains a hex dump of the entire packet (Wireshark, 2014)



In my first run, I started seeing some communication with 74.125.22.100. I also noticed some retransmission packets and packets containing the following messages:

destination host is unreachable (host administratively prohibited)

They appeared to be only for ICMP packets, so I imagine they must be blocking those packets using a firewall or other network policy.

Some further investigation provided additional information:

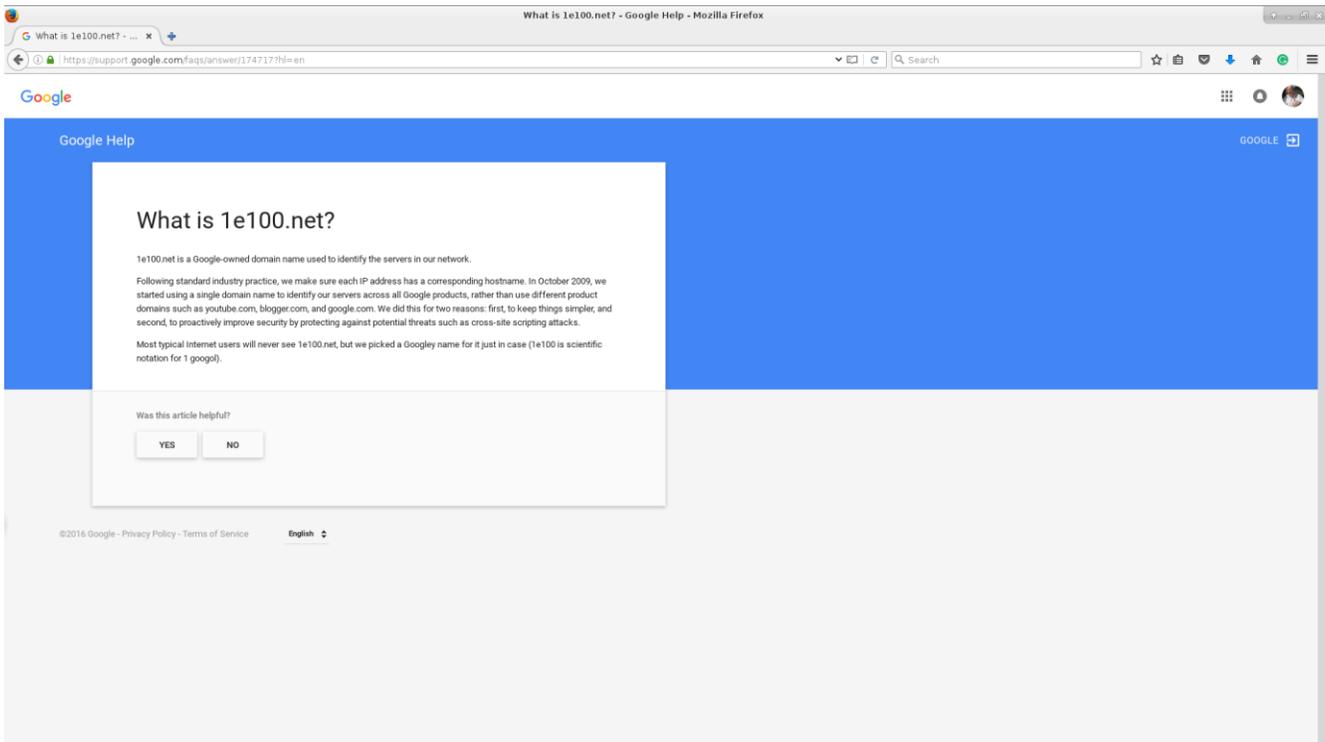
```
Terminal - root@biff:/home/biff/docs/UticaCollege/wiresharklab
File Edit View Terminal Tabs Help
root@centos7:~/var/www/html/drop/UticaCollege
root@biff:/home/biff/docs/UticaCollege/wiresharklab
[root@biff wiresharklab]# host 74.125.22.100
100.22.125.74.in-addr.arpa domain name pointer qh-in-f100.1e100.net.
[root@biff wiresharklab]# nmap qh-in-f100.1e100.net

Starting Nmap 6.40 ( http://nmap.org ) at 2016-09-29 19:44 EDT
Nmap scan report for qh-in-f100.1e100.net (74.125.22.100)
Host is up (0.021s latency).
Not shown: 996 filtered ports
PORT      STATE SERVICE
80/tcp    open  http
443/tcp   open  https
554/tcp   open  rtsp
7070/tcp  open  realserver

Nmap done: 1 IP address (1 host up) scanned in 5.08 seconds
[root@biff wiresharklab]#
```

It appears that this is Google:

<https://support.google.com/faqs/answer/174717?hl=en>

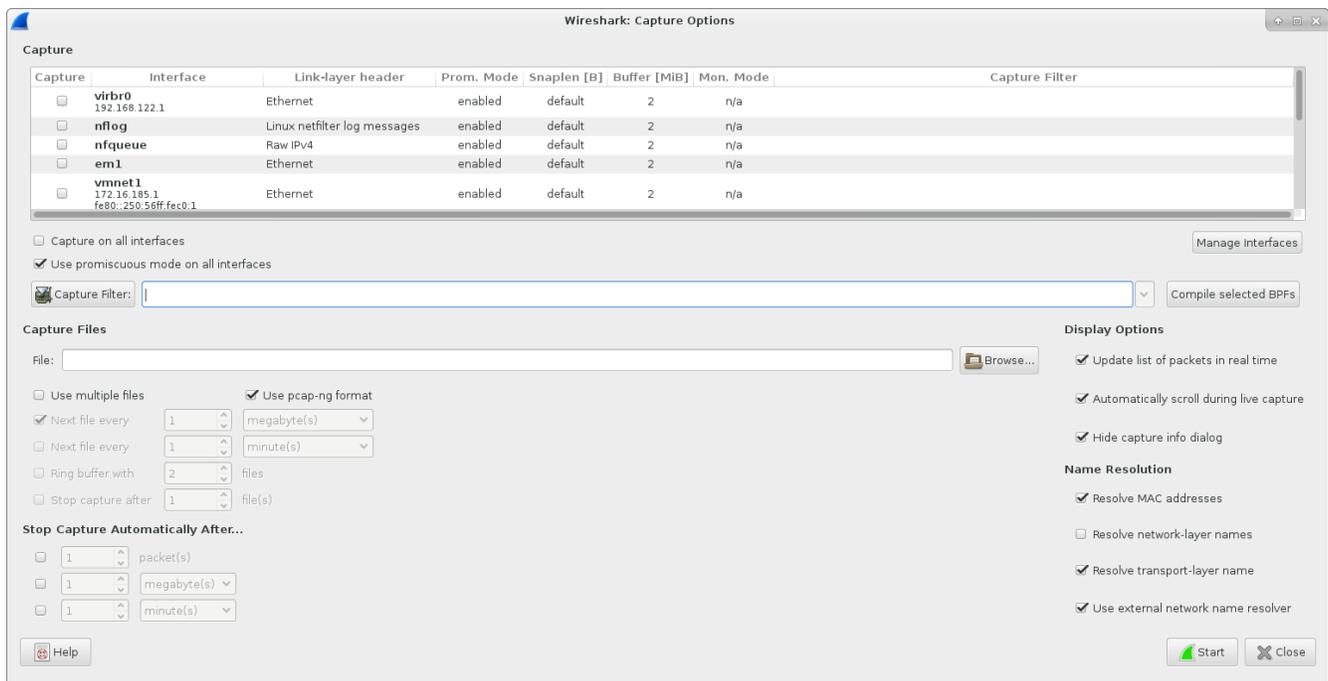


The lab then asks us to select the STOP button and describe the results in your lab:

The screenshot shows the Wireshark interface with the following details:

- Filter:** Expression... Clear Apply Save
- Packet List:**

No.	Time	Source	Destination	Protocol	Length	Info
45680	5186.741496729	192.168.1.6	173.194.68.139	TLv1.2	100	Application Data
45681	5186.741511280	192.168.1.6	173.194.68.139	TLv1.2	100	[TCP Retransmission] Application Data
45682	5186.742290065	192.168.1.6	173.194.68.139	TLv1.2	85	Encrypted Alert
45683	5186.742307409	192.168.1.6	173.194.68.139	TLv1.2	85	[TCP Retransmission] Encrypted Alert
45684	5186.742497697	192.168.1.6	173.194.68.139	TCP	54	rbakcup2 > https [FIN, ACK] Seq=2772 Ack=1037 Win=5337 Len=0
45685	5186.742511159	192.168.1.6	173.194.68.139	TCP	54	[TCP Out-Of-Order] rbakcup2 > https [FIN, ACK] Seq=2772 Ack=1037 Win=5337 Len=0
45686	5186.767945588	173.194.68.139	192.168.1.6	TCP	60	https > rbakcup2 [RST] Seq=1037 Win=0 Len=0
45687	5186.768067452	192.168.1.8	173.194.68.139	ICMP	62	Destination unreachable (Host administratively prohibited)
45688	5187.421418004	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover - Transaction ID 0x86e88039
45689	5187.761902007	Actionte_20:7a:f7	Broadcast	ARP	60	Who has 192.168.1.10? Tell 192.168.1.1
45690	5187.926244638	192.168.1.6	192.168.1.255	BROWSER	243	Host Announcement BIFFSOCK:3MKDZJ, Workstation, Server, NT Workstation
45691	5187.926285297	192.168.1.6	192.168.1.255	BROWSER	243	Host Announcement BIFFSOCK:3MKDZJ, Workstation, Server, NT Workstation
45692	5188.034970196	Actionte_20:7a:f7	Spanning-tree (for-brstp	60	Conf. Root = 32768/0/00:7f:2b:20:7a:f8 Cost = 0 Port = 0x8004	
45693	5188.750418531	Actionte_20:7a:f7	Broadcast	ARP	60	Who has 192.168.1.10? Tell 192.168.1.1
45694	5188.774016766	Actionte_20:7a:f7	Broadcast	ARP	60	Who has 192.168.1.10? Tell 192.168.1.1
45695	5189.048642833	173.194.206.189	192.168.1.8	TLv1.2	127	Application Data
45696	5189.948748109	192.168.1.8	173.194.206.189	TCP	66	50738 > https [ACK] Seq=3386 Ack=4609 Win=33536 Len=0 TSval=1540417528 TSecr=714311826
45697	5189.970152493	Actionte_20:7a:f7	Spanning-tree (for-brstp	60	Conf. Root = 32768/0/00:7f:2b:20:7a:f8 Cost = 0 Port = 0x8004	
45698	5190.354374779	192.168.1.8	192.168.1.1	DNS	75	Standard query 0xad9 A play.google.com
45699	5190.354408368	192.168.1.8	192.168.1.1	DNS	75	Standard query 0xad9 AAAA play.google.com
45700	5190.360754865	192.168.1.8	172.217.2.14	TLv1.2	1451	Application Data
45701	5190.365661159	192.168.1.8	172.217.2.14	TLv1.2	369	Application Data
45702	5190.375181237	172.217.2.14	192.168.1.8	TCP	66	https > 37070 [ACK] Seq=13486 Ack=2912 Win=50944 Len=0 TSval=228467238 TSecr=1540417940
45703	5190.375854320	192.168.1.1	192.168.1.8	DNS	182	Standard query response 0xad9 CNAME play.l.google.com A 173.194.204.101 A 173.194.204.102 A 173.194.204.103 A 173.194.204.104 A 173.194.204.105 A 173.194.204.106 A 173.194.204.107 A 173.194.204.108 A 173.194.204.109 A 173.194.204.110 A 173.194.204.111 A 173.194.204.112 A 173.194.204.113 A 173.194.204.114 A 173.194.204.115 A 173.194.204.116 A 173.194.204.117 A 173.194.204.118 A 173.194.204.119 A 173.194.204.120 A 173.194.204.121 A 173.194.204.122 A 173.194.204.123 A 173.194.204.124 A 173.194.204.125 A 173.194.204.126 A 173.194.204.127 A 173.194.204.128 A 173.194.204.129 A 173.194.204.130 A 173.194.204.131 A 173.194.204.132 A 173.194.204.133 A 173.194.204.134 A 173.194.204.135 A 173.194.204.136 A 173.194.204.137 A 173.194.204.138 A 173.194.204.139 A 173.194.204.140 A 173.194.204.141 A 173.194.204.142 A 173.194.204.143 A 173.194.204.144 A 173.194.204.145 A 173.194.204.146 A 173.194.204.147 A 173.194.204.148 A 173.194.204.149 A 173.194.204.150 A 173.194.204.151 A 173.194.204.152 A 173.194.204.153 A 173.194.204.154 A 173.194.204.155 A 173.194.204.156 A 173.194.204.157 A 173.194.204.158 A 173.194.204.159 A 173.194.204.160 A 173.194.204.161 A 173.194.204.162 A 173.194.204.163 A 173.194.204.164 A 173.194.204.165 A 173.194.204.166 A 173.194.204.167 A 173.194.204.168 A 173.194.204.169 A 173.194.204.170 A 173.194.204.171 A 173.194.204.172 A 173.194.204.173 A 173.194.204.174 A 173.194.204.175 A 173.194.204.176 A 173.194.204.177 A 173.194.204.178 A 173.194.204.179 A 173.194.204.180 A 173.194.204.181 A 173.194.204.182 A 173.194.204.183 A 173.194.204.184 A 173.194.204.185 A 173.194.204.186 A 173.194.204.187 A 173.194.204.188 A 173.194.204.189 A 173.194.204.190 A 173.194.204.191 A 173.194.204.192 A 173.194.204.193 A 173.194.204.194 A 173.194.204.195 A 173.194.204.196 A 173.194.204.197 A 173.194.204.198 A 173.194.204.199 A 173.194.204.200 A 173.194.204.201 A 173.194.204.202 A 173.194.204.203 A 173.194.204.204 A 173.194.204.205 A 173.194.204.206 A 173.194.204.207 A 173.194.204.208 A 173.194.204.209 A 173.194.204.210 A 173.194.204.211 A 173.194.204.212 A 173.194.204.213 A 173.194.204.214 A 173.194.204.215 A 173.194.204.216 A 173.194.204.217 A 173.194.204.218 A 173.194.204.219 A 173.194.204.220 A 173.194.204.221 A 173.194.204.222 A 173.194.204.223 A 173.194.204.224 A 173.194.204.225 A 173.194.204.226 A 173.194.204.227 A 173.194.204.228 A 173.194.204.229 A 173.194.204.230 A 173.194.204.231 A 173.194.204.232 A 173.194.204.233 A 173.194.204.234 A 173.194.204.235 A 173.194.204.236 A 173.194.204.237 A 173.194.204.238 A 173.194.204.239 A 173.194.204.240 A 173.194.204.241 A 173.194.204.242 A 173.194.204.243 A 173.194.204.244 A 173.194.204.245 A 173.194.204.246 A 173.194.204.247 A 173.194.204.248 A 173.194.204.249 A 173.194.204.250 A 173.194.204.251 A 173.194.204.252 A 173.194.204.253 A 173.194.204.254 A 173.194.204.255 A 173.194.204.256 A 173.194.204.257 A 173.194.204.258 A 173.194.204.259 A 173.194.204.260 A 173.194.204.261 A 173.194.204.262 A 173.194.204.263 A 173.194.204.264 A 173.194.204.265 A 173.194.204.266 A 173.194.204.267 A 173.194.204.268 A 173.194.204.269 A 173.194.204.270 A 173.194.204.271 A 173.194.204.272 A 173.194.204.273 A 173.194.204.274 A 173.194.204.275 A 173.194.204.276 A 173.194.204.277 A 173.194.204.278 A 173.194.204.279 A 173.194.204.280 A 173.194.204.281 A 173.194.204.282 A 173.194.204.283 A 173.194.204.284 A 173.194.204.285 A 173.194.204.286 A 173.194.204.287 A 173.194.204.288 A 173.194.204.289 A 173.194.204.290 A 173.194.204.291 A 173.194.204.292 A 173.194.204.293 A 173.194.204.294 A 173.194.204.295 A 173.194.204.296 A 173.194.204.297 A 173.194.204.298 A 173.194.204.299 A 173.194.204.300 A 173.194.204.301 A 173.194.204.302 A 173.194.204.303 A 173.194.204.304 A 173.194.204.305 A 173.194.204.306 A 173.194.204.307 A 173.194.204.308 A 173.194.204.309 A 173.194.204.310 A 173.194.204.311 A 173.194.204.312 A 173.194.204.313 A 173.194.204.314 A 173.194.204.315 A 173.194.204.316 A 173.194.204.317 A 173.194.204.318 A 173.194.204.319 A 173.194.204.320 A 173.194.204.321 A 173.194.204.322 A 173.194.204.323 A 173.194.204.324 A 173.194.204.325 A 173.194.204.326 A 173.194.204.327 A 173.194.204.328 A 173.194.204.329 A 173.194.204.330 A 173.194.204.331 A 173.194.204.332 A 173.194.204.333 A 173.194.204.334 A 173.194.204.335 A 173.194.204.336 A 173.194.204.337 A 173.194.204.338 A 173.194.204.339 A 173.194.204.340 A 173.194.204.341 A 173.194.204.342 A 173.194.204.343 A 173.194.204.344 A 173.194.204.345 A 173.194.204.346 A 173.194.204.347 A 173.194.204.348 A 173.194.204.349 A 173.194.204.350 A 173.194.204.351 A 173.194.204.352 A 173.194.204.353 A 173.194.204.354 A 173.194.204.355 A 173.194.204.356 A 173.194.204.357 A 173.194.204.358 A 173.194.204.359 A 173.194.204.360 A 173.194.204.361 A 173.194.204.362 A 173.194.204.363 A 173.194.204.364 A 173.194.204.365 A 173.194.204.366 A 173.194.204.367 A 173.194.204.368 A 173.194.204.369 A 173.194.204.370 A 173.194.204.371 A 173.194.204.372 A 173.194.204.373 A 173.194.204.374 A 173.194.204.375 A 173.194.204.376 A 173.194.204.377 A 173.194.204.378 A 173.194.204.379 A 173.194.204.380 A 173.194.204.381 A 173.194.204.382 A 173.194.204.383 A 173.194.204.384 A 173.194.204.385 A 173.194.204.386 A 173.194.204.387 A 173.194.204.388 A 173.194.204.389 A 173.194.204.390 A 173.194.204.391 A 173.194.204.392 A 173.194.204.393 A 173.194.204.394 A 173.194.204.395 A 173.194.204.396 A 173.194.204.397 A 173.194.204.398 A 173.194.204.399 A 173.194.204.400 A 173.194.204.401 A 173.194.204.402 A 173.194.204.403 A 173.194.204.404 A 173.194.204.405 A 173.194.204.406 A 173.194.204.407 A 173.194.204.408 A 173.194.204.409 A 173.194.204.410 A 173.194.204.411 A 173.194.204.412 A 173.194.204.413 A 173.194.204.414 A 173.194.204.415 A 173.194.204.416 A 173.194.204.417 A 173.194.204.418 A 173.194.204.419 A 173.194.204.420 A 173.194.204.421 A 173.194.204.422 A 173.194.204.423 A 173.194.204.424 A 173.194.204.425 A 173.194.204.426 A 173.194.204.427 A 173.194.204.428 A 173.194.204.429 A 173.194.204.430 A 173.194.204.431 A 173.194.204.432 A 173.194.204.433 A 173.194.204.434 A 173.194.204.435 A 173.194.204.436 A 173.194.204.437 A 173.194.204.438 A 173.194.204.439 A 173.194.204.440 A 173.194.204.441 A 173.194.204.442 A 173.194.204.443 A 173.194.204.444 A 173.194.204.445 A 173.194.204.446 A 173.194.204.447 A 173.194.204.448 A 173.194.204.449 A 173.194.204.450 A 173.194.204.451 A 173.194.204.452 A 173.194.204.453 A 173.194.204.454 A 173.194.204.455 A 173.194.204.456 A 173.194.204.457 A 173.194.204.458 A 173.194.204.459 A 173.194.204.460 A 173.194.204.461 A 173.194.204.462 A 173.194.204.463 A 173.194.204.464 A 173.194.204.465 A 173.194.204.466 A 173.194.204.467 A 173.194.204.468 A 173.194.204.469 A 173.194.204.470 A 173.194.204.471 A 173.194.204.472 A 173.194.204.473 A 173.194.204.474 A 173.194.204.475 A 173.194.204.476 A 173.194.204.477 A 173.194.204.478 A 173.194.204.479 A 173.194.204.480 A 173.194.204.481 A 173.194.204.482 A 173.194.204.483 A 173.194.204.484 A 173.194.204.485 A 173.194.204.486 A 173.194.204.487 A 173.194.204.488 A 173.194.204.489 A 173.194.204.490 A 173.194.204.491 A 173.194.204.492 A 173.194.204.493 A 173.194.204.494 A 173.194.204.495 A 173.194.204.496 A 173.194.204.497 A 173.194.204.498 A 173.194.204.499 A 173.194.204.500 A 173.194.204.501 A 173.194.204.502 A 173.194.204.503 A 173.194.204.504 A 173.194.204.505 A 173.194.204.506 A 173.194.204.507 A 173.194.204.508 A 173.194.204.509 A 173.194.204.510 A 173.194.204.511 A 173.194.204.512 A 173.194.204.513 A 173.194.204.514 A 173.194.204.515 A 173.194.204.516 A 173.194.204.517 A 173.194.204.518 A 173.194.204.519 A 173.194.204.520 A 173.194.204.521 A 173.194.204.522 A 173.194.204.523 A 173.194.204.524 A 173.194.204.525 A 173.194.204.526 A 173.194.204.527 A 173.194.204.528 A 173.194.204.529 A 173.194.204.530 A 173.194.204.531 A 173.194.204.532 A 173.194.204.533 A 173.194.204.534 A 173.194.204.535 A 173.194.204.536 A 173.194.204.537 A 173.194.204.538 A 173.194.204.539 A 173.194.204.540 A 173.194.204.541 A 173.194.204.542 A 173.194.204.543 A 173.194.204.544 A 173.194.204.545 A 173.194.204.546 A 173.194.204.547 A 173.194.204.548 A 173.194.204.549 A 173.194.204.550 A 173.194.204.551 A 173.194.204.552 A 173.194.204.553 A 173.194.204.554 A 173.194.204.555 A 173.194.204.556 A 173.194.204.557 A 173.194.204.558 A 173.194.204.559 A 173.194.204.560 A 173.194.204.561 A 173.194.204.562 A 173.194.204.563 A 173.194.204.564 A 173.194.204.565 A 173.194.204.566 A 173.194.204.567 A 173.194.204.568 A 173.194.204.569 A 173.194.204.570 A 173.194.204.571 A 173.194.204.572 A 173.194.204.573 A 173.194.204.574 A 173.194.204.575 A 173.194.204.576 A 173.194.204.577 A 173.194.204.578 A 173.194.204.579 A 173.194.204.580 A 173.194.204.581 A 173.194.204.582 A 173.194.204.583 A 173.194.204.584 A 173.194.204.585 A 173.194.204.586 A 173.194.204.587 A 173.194.204.588 A 173.194.204.589 A 173.194.204.590 A 173.194.204.591 A 173.194.204.592 A 173.194.204.593 A 173.194.204.594 A 173.194.204.595 A 173.194.204.596 A 173.194.204.597 A 173.194.204.598 A 173.194.204.599 A 173.194.204.600 A 173.194.204.601 A 173.194.204.602 A 173.194.204.603 A 173.194.204.604 A 173.194.204.605 A 173.194.204.606 A 173.194.204.607 A 173.194.204.608 A 173.194.204.609 A 173.194.204.610 A 173.194.204.611 A 173.194.204.612 A 173.194.204.613 A 173.194.204.614 A 173.194.204.615 A 173.194.204.616 A 173.194.204.617 A 173.194.204.618 A 173.194.204.619 A 173.194.204.620 A 173.194.204.621 A 173.194.204.622 A 173.194.204.623 A 173.194.204.624 A 173.194.204.625 A 173.194.204.626 A 173.194.204.627 A 173.194.204.628 A 173.194.204.629 A 173.194.204.630 A 173.194.204.631 A 173.194.204.632 A 173.194.204.633 A 173.194.204.634 A 173.194.204.635 A 173.194.204.636 A 173.194.204.637 A 173.194.204.638 A 173.194.204.639 A 173.194.204.640 A 173.194.204.641 A 173.194.204.642 A 173.194.204.643 A 173.194.204.644 A 173.194.204.645 A 173.194.204.64



I played with turning on/off those options. The results were intended to show that packets were updated in the packet frame window in real-time (no buffering) as well as enabling scrolling using the scroll bar on the right side of the screen.

Filter Packets with the Filter Bar

Our lab now asks us to filter packets. Without filtering, it's difficult to pinpoint what exactly we're looking for because of the glut of information. By narrowing down our search scope, we can more easily find specific things. The GUI contains a "Filter" bar that allows for expressions.

Filtering by IP:

Filter: ip

No.	Time	Source	Destination	Protoc	Length	Info
191579	12.068	192.168.1.1	192.168.1.8	DNS	115	Standard query response 0x3b70 AAAA 2607:f800:400d:c061:bd
191598	12.096	65.55.56.97	74.125.142.178	TCP	54	[TCP Keep-Alive] 54364 > http [ACK] Seq=1395 Ack=442 Win=30336 Len=0
191600	12.097	65.55.56.97	74.125.142.178	TCP	53	[TCP Keep-Alive ACK] http > 54364 [ACK] Seq=442 Ack=1395 Win=32256 Len=0
191603	12.097	835649335	192.168.1.8	TLSv1.2	127	Application Data
191603	12.097	835649335	192.168.1.8	TCP	66	50338 > https [ACK] Seq=2601 Ack=3751 Win=34560 Len=0 TSval=1554376664 TSecr=702342563
191612	12.106	707576959	74.125.142.178	TCP	54	[TCP Keep-Alive] 54364 > http [ACK] Seq=1395 Ack=442 Win=30336 Len=0
191613	12.106	72038943	74.125.142.178	TCP	53	[TCP Keep-Alive ACK] http > 54364 [ACK] Seq=442 Ack=1395 Win=32256 Len=0
191613	12.109	130458729	192.168.1.8	TCP	143	Destination unreachable (Host administratively prohibited)
191618	12.109	32652367	192.168.1.6	TCP	54	4-tieropgw > https [ACK] Seq=7180 Ack=15927 Win=64242 Len=0
191619	12.109	32652367	192.168.1.6	TCP	54	[TCP Dup ACK 191618#1] 4-tieropgw > https [ACK] Seq=7180 Ack=15927 Win=64242 Len=0
191622	12.111	602658527	192.168.1.1	SSDP	369	NOTIFY * HTTP/1.1
191624	12.111	605430318	192.168.1.1	SSDP	369	NOTIFY * HTTP/1.1
191625	12.111	605430318	192.168.1.1	SSDP	369	NOTIFY * HTTP/1.1
191626	12.111	606415584	192.168.1.1	SSDP	441	NOTIFY * HTTP/1.1
191627	12.111	609508452	192.168.1.1	SSDP	441	NOTIFY * HTTP/1.1
191628	12.111	607084203	192.168.1.1	SSDP	441	NOTIFY * HTTP/1.1
191629	12.111	607481155	192.168.1.1	SSDP	378	NOTIFY * HTTP/1.1
191630	12.111	607583062	192.168.1.1	SSDP	378	NOTIFY * HTTP/1.1
191631	12.111	608123658	192.168.1.1	SSDP	378	NOTIFY * HTTP/1.1
191632	12.111	609548335	192.168.1.1	SSDP	421	NOTIFY * HTTP/1.1
191633	12.111	60969176	192.168.1.1	SSDP	421	NOTIFY * HTTP/1.1
191634	12.111	615895771	192.168.1.1	SSDP	421	NOTIFY * HTTP/1.1
191635	12.111	615947369	192.168.1.1	SSDP	433	NOTIFY * HTTP/1.1
191636	12.111	615969871	192.168.1.1	SSDP	433	NOTIFY * HTTP/1.1
191637	12.111	615991527	192.168.1.1	SSDP	433	NOTIFY * HTTP/1.1
191638	12.111	616020988	192.168.1.1	SSDP	417	NOTIFY * HTTP/1.1
191639	12.111	616047609	192.168.1.1	SSDP	417	NOTIFY * HTTP/1.1
191640	12.111	616061420	192.168.1.1	SSDP	417	NOTIFY * HTTP/1.1
191641	12.111	616078793	192.168.1.1	SSDP	378	NOTIFY * HTTP/1.1
191642	12.111	616092470	192.168.1.1	SSDP	378	NOTIFY * HTTP/1.1
191643	12.111	616105671	192.168.1.1	SSDP	378	NOTIFY * HTTP/1.1
191644	12.111	616378071	192.168.1.1	SSDP	435	NOTIFY * HTTP/1.1
191645	12.111	616416910	192.168.1.1	SSDP	435	NOTIFY * HTTP/1.1
191646	12.111	616436257	192.168.1.1	SSDP	435	NOTIFY * HTTP/1.1
191647	12.111	616450291	192.168.1.1	SSDP	435	NOTIFY * HTTP/1.1
191648	12.111	616463963	192.168.1.1	SSDP	435	NOTIFY * HTTP/1.1
191649	12.111	616478011	192.168.1.1	SSDP	435	NOTIFY * HTTP/1.1

Frame 191579: 115 bytes on wire (920 bits), 115 bytes captured (920 bits) on interface 0
 Ethernet II, Src: Actionte_20:7a:f7 (00:7f:2b:20:7a:f7), Dst: IntelCor_7b:7e:64 (aa:8b:b4:7b:7e:64)
 Internet Protocol Version 4, Src: 192.168.1.1 (192.168.1.1), Dst: 192.168.1.8 (192.168.1.8)
 User Datagram Protocol, Src Port: domain (53), Dst Port: 52874 (52874)
 Domain Name System (response)

```

0000  a0 8b b4 7b 7e 64 00 7f 2b 20 7a f7 08 00 45 00  ...(.d.. (.z...E.
0010  00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ..q.N.. .z.B...
0020  01 08 01 0b c4 e2 11 c8 e5 e4 8d 8d 01 25 80 18  .... . . . . .
0030  00 01 04 e2 5f 00 00 01 01 08 0a 29 e5 44 46 5c 5d  .... . . . . .) .DP.
0040  de a5 17 c3 03 00 38 00 00 00 00 00 00 00 00 4f  ea ..... . . . . .
0050  d8 c8 2b 5f 85 e6 9f 7b 5a b0 d6 b1 29 92 11 ea  .... . . . . .
0060  12 1e 63 8d e7 2f 1e e2 6e a1 7c 0d 54 ae 98 cd  .... . . . . .) .n .T . . .
0070  01 94 fb 9b 35 48 ca 3e 0b f9 44 da 50 c3 17  .... . . . . .) .D .P . .
    
```

I show that I am only capturing IP packets.

Filtering by TCP:

Filter: tcp

No.	Time	Source	Destination	Protoc	Length	Info
602	165.2305956	172.217.2.14	192.168.1.8	TLSv1.2	328	New Session Ticket, Change Cipher Spec, Hello Request, Hello Request
603	165.2304242	172.217.2.14	192.168.1.8	TLSv1.2	128	Application Data
604	165.2304529	172.217.2.14	192.168.1.8	TLSv1.2	108	Application Data
605	165.2305097	172.217.2.14	192.168.1.8	TLSv1.2	104	Application Data
606	165.2307678	192.168.1.8	172.217.2.14	TLSv1.2	104	Application Data
607	165.2463929	172.217.2.14	192.168.1.8	TCP	66	https > 54282 [ACK] Seq=4388 Ack=2232 Win=49408 Len=0 TSval=123096069 TSecr=1555086737
608	165.3163411	173.194.208.189	192.168.1.8	TCP	66	https > 54282 [ACK] Seq=343 Ack=1074 Win=287 Len=0 TSval=1555086836 TSecr=898604467
609	165.3164578	173.194.208.189	192.168.1.6	TCP	66	44598 > https [ACK] Seq=343 Ack=1074 Win=287 Len=0 TSval=1555086836 TSecr=898604467
610	165.3165078	173.194.208.189	192.168.1.6	TCP	545	Application Data
611	165.3165756	192.168.1.8	74.125.22.189	TCP	573	Destination unreachable (Host administratively prohibited)
612	165.3257268	172.217.2.14	192.168.1.8	TLSv1.2	389	Application Data
613	165.3258055	172.217.2.14	192.168.1.8	TLSv1.2	327	Application Data
614	165.3258301	172.217.2.14	192.168.1.8	TCP	112	Application Data
615	165.3263693	192.168.1.8	172.217.2.14	TCP	66	54282 > https [ACK] Seq=2232 Ack=5218 Win=46336 Len=0 TSval=1555086846 TSecr=123096149
616	165.3264706	192.168.1.8	172.217.2.14	TLSv1.2	112	Application Data
617	165.3627386	172.217.2.14	192.168.1.8	TCP	66	https > 54282 [ACK] Seq=5218 Ack=2278 Win=49408 Len=0 TSval=123096206 TSecr=1555086846
618	165.3627972	173.194.208.189	192.168.1.8	TLSv1.2	506	Application Data
619	165.3628194	192.168.1.8	173.194.208.189	TCP	66	44598 > https [ACK] Seq=343 Ack=1514 Win=296 Len=0 TSval=1555086902 TSecr=898604534
620	165.3628461	174.125.22.189	192.168.1.6	TLSv1.2	494	Application Data
621	165.3630001	192.168.1.8	74.125.22.189	TCP	573	Destination unreachable (Host administratively prohibited)
622	165.3630356	192.168.1.6	74.125.22.189	TCP	54	natplan > https [ACK] Seq=1 Ack=1298 Win=65535 Len=0
623	165.3630478	192.168.1.6	74.125.22.189	TCP	54	[TCP Dup ACK 622#1] natplan > https [ACK] Seq=1 Ack=1298 Win=65535 Len=0

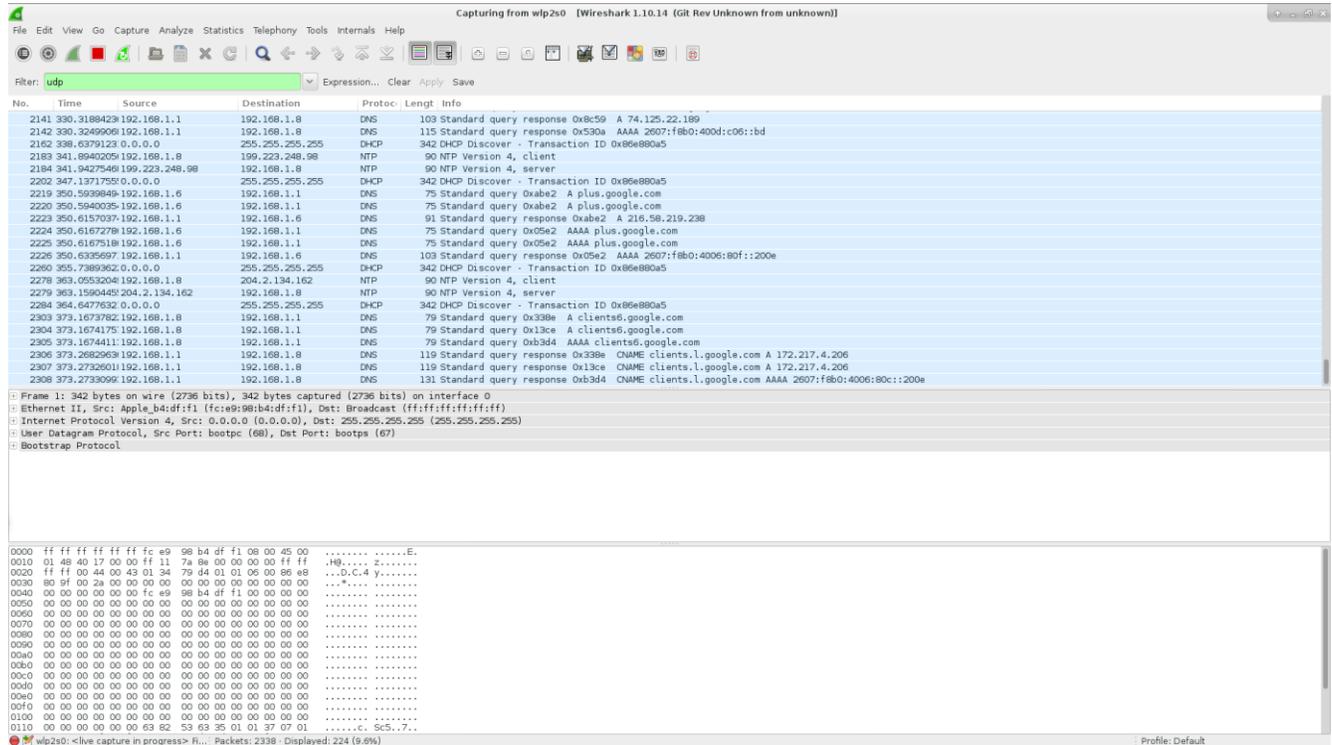
Frame 9: 127 bytes on wire (1016 bits), 127 bytes captured (1016 bits) on interface 0
 Ethernet II, Src: Actionte_20:7a:f7 (00:7f:2b:20:7a:f7), Dst: IntelCor_7b:7e:64 (aa:8b:b4:7b:7e:64)
 Internet Protocol Version 4, Src: 173.194.66.189 (173.194.66.189), Dst: 192.168.1.8 (192.168.1.8)
 Transmission Control Protocol, Src Port: https (443), Dst Port: 50338 (50338), Seq: 1, Ack: 1, Len: 61
 Secure Sockets Layer

```

0000  a0 8b b4 7b 7e 64 00 7f 2b 20 7a f7 08 00 45 00  ...(.d.. (.z...E.
0010  00 71 ac 25 00 00 30 06 8c 32 ad c2 42 bd cd a8  ..q.N.. .z.B...
0020  01 08 01 0b c4 e2 11 c8 e5 e4 8d 8d 01 25 80 18  .... . . . . .
0030  00 01 04 e2 5f 00 00 01 01 08 0a 29 e5 44 46 5c 5d  .... . . . . .) .DP.
0040  de a5 17 c3 03 00 38 00 00 00 00 00 00 00 00 4f  ea ..... . . . . .
0050  d8 c8 2b 5f 85 e6 9f 7b 5a b0 d6 b1 29 92 11 ea  .... . . . . .
0060  12 1e 63 8d e7 2f 1e e2 6e a1 7c 0d 54 ae 98 cd  .... . . . . .) .n .T . . .
0070  01 94 fb 9b 35 48 ca 3e 0b f9 44 da 50 c3 17  .... . . . . .) .D .P . .
    
```

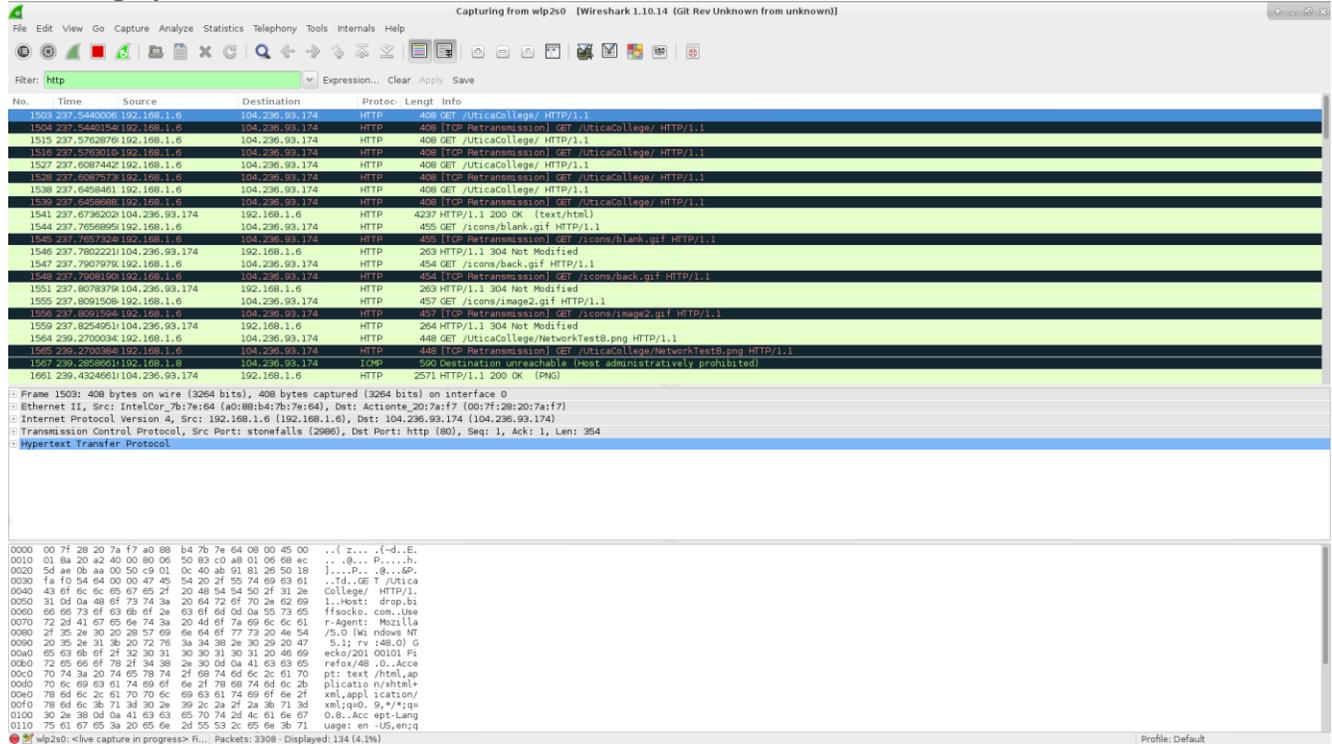
Filtering by TCP shows application layer data – perhaps the most interesting things I’m looking for from a cyber security perspective.

Filtering by UDP:



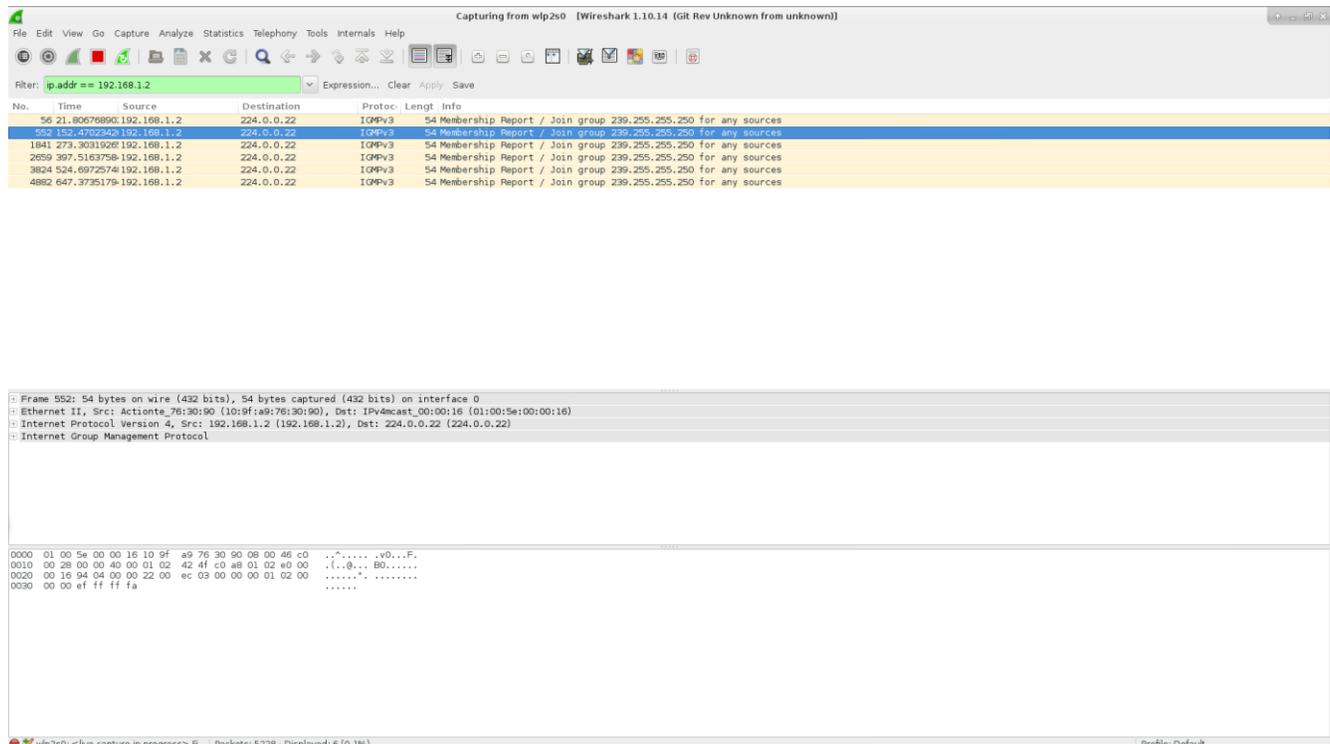
Filtering by UDP shows the expected UDP applications (NTP, DNS, DHCP, NFS etc).

Filtering by HTTP:



Filtering by “http” shows mostly http traffic. I did encounter an ICMP packet with this filter enabled, which was kind of weird. I have no explanation for this.

Specifying an IP address:



When filtering by IP address, I narrow down the packets that have as a source or destination the specified IP address.

View Packet Summaries with the Packet List Window

The packet frame in the below picture shows a number of columns.

Column 1 – shows the packet number

Column 2 - shows the time in a number of formats. In the example below, I have chosen the number of seconds since the live capture began. It is accurate to the nanosecond.

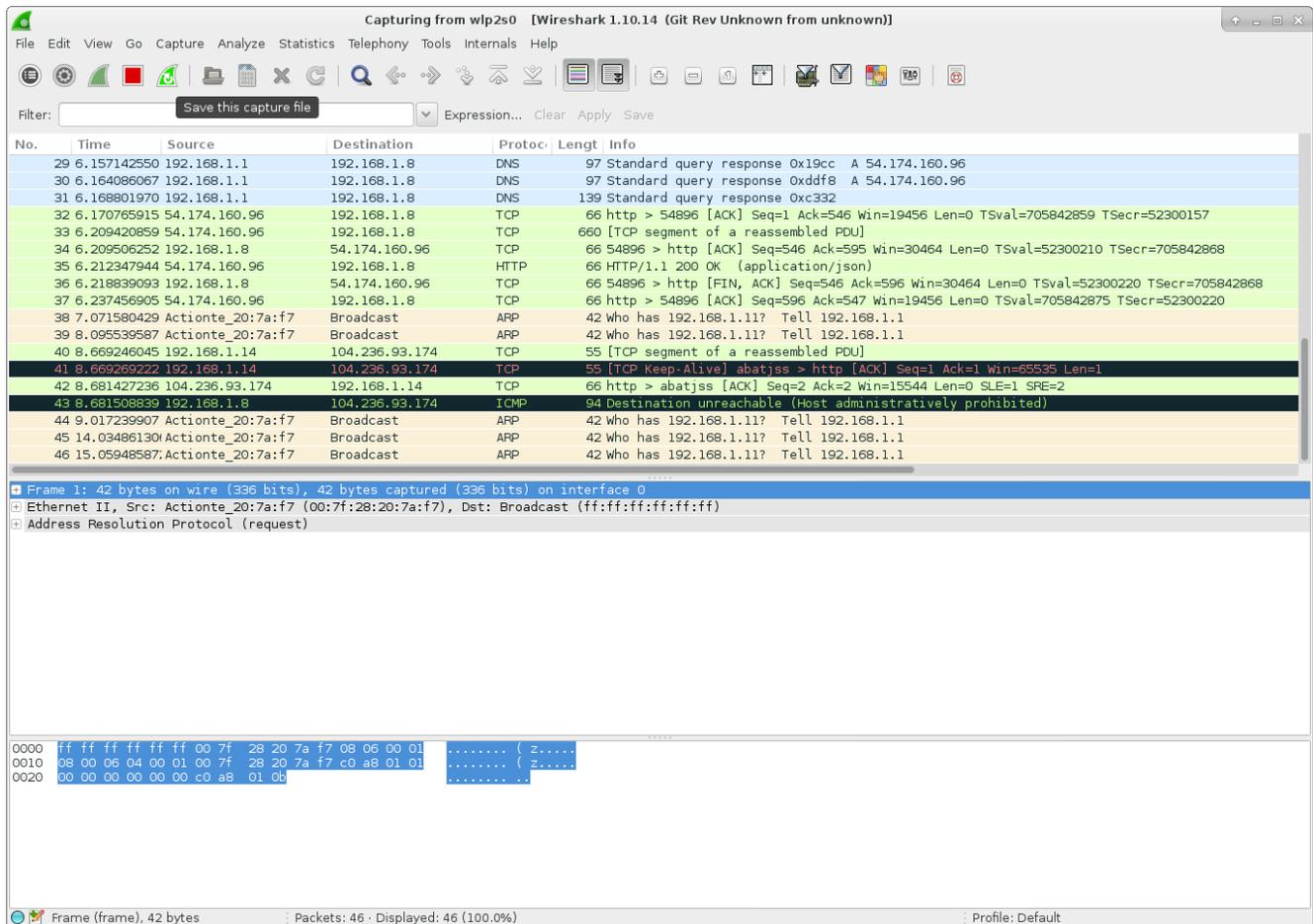
Column 3 - is the source IP address (where the packet is coming from)

Column 4 - is the destination IP address (where the packet is going to)

Column 5 - is the protocol

Column 6 - is the length of the packet

Column 7 - is information associated with the packet.



Study Packet Details with the Packet Details Window

By highlighting a packet in the packet summary window, you can view further information about that packet in the packet details frame.

Here are some of the details of an ARP packet:

The image shows the Wireshark interface with a packet capture list at the top and a detailed view of frame 31301 below. The packet list shows several ARP and ICMPv6 packets. The selected frame 31301 is an ARP request from Actionte_20:7a:f7 to a broadcast destination. The details pane shows the Ethernet II, Destination, Source, and ARP protocol layers. The hex dump at the bottom shows the raw bytes of the frame.

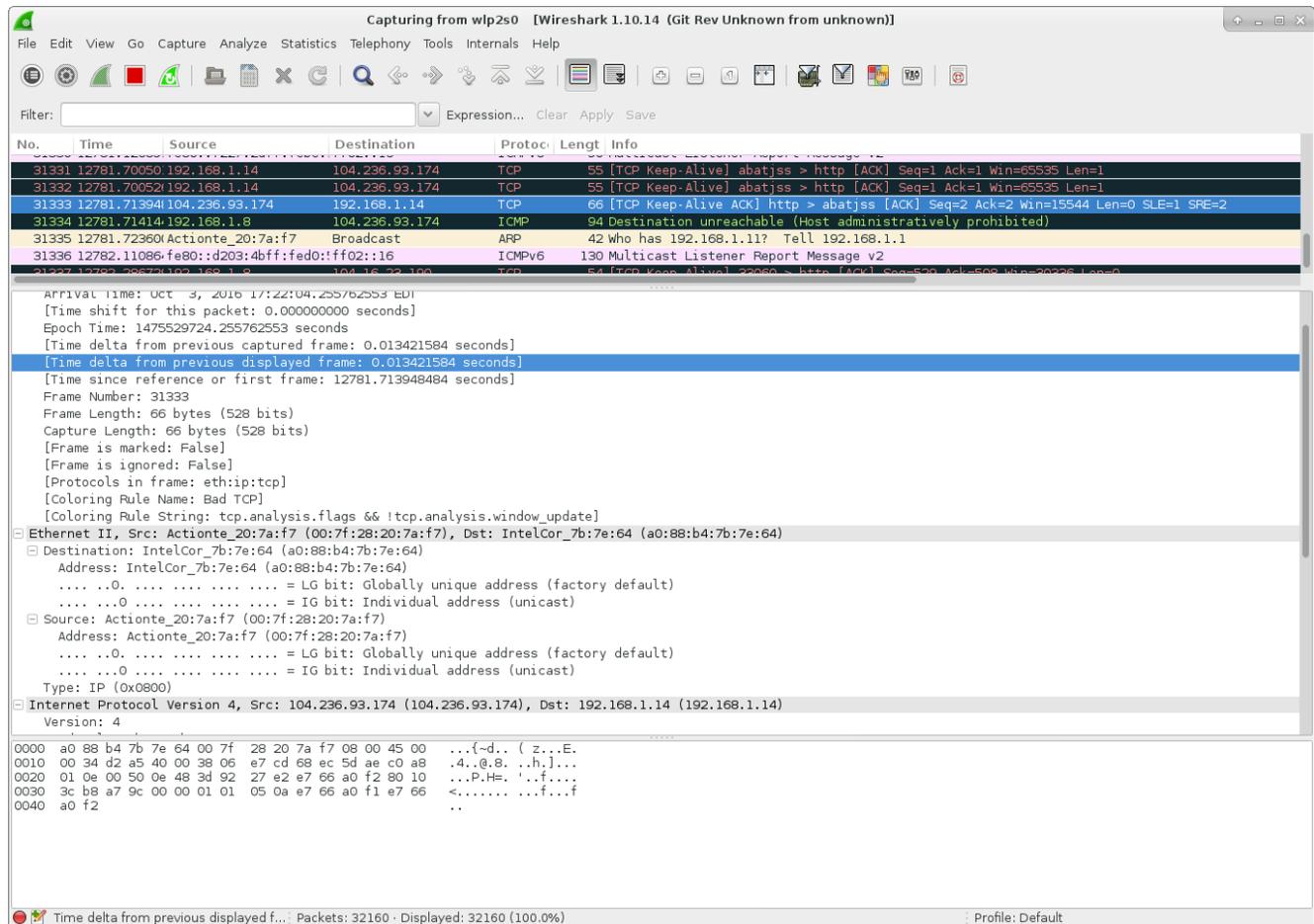
No.	Time	Source	Destination	Protoc.	Length	Info
31321	12774.73351	Actionte_20:7a:f7	Broadcast	ARP	42	Who has 192.168.1.11? Tell 192.168.1.1
31322	12777.27282	Actionte_20:7a:f7	IntelCor_7b:7e:64	ARP	42	Who has 192.168.1.8? Tell 192.168.1.1
31323	12777.27288	IntelCor_7b:7e:64	Actionte_20:7a:f7	ARP	42	192.168.1.8 is at a0:88:b4:7b:7e:64
31324	12779.72358	Actionte_20:7a:f7	Broadcast	ARP	42	Who has 192.168.1.11? Tell 192.168.1.1
31325	12780.72348	Actionte_20:7a:f7	Broadcast	ARP	42	Who has 192.168.1.11? Tell 192.168.1.1
31326	12781.04614	fe80::129f:a9ff:fe76::ff02::1		ICMPv6	90	Multicast Listener Query
31327	12781.05266	fe80::a288:b4ff:fe7b::ff02::16		ICMPv6	90	Multicast Listener Report Message v2

Frame 31301: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface 0

- Interface id: 0
- Encapsulation type: Ethernet (1)
- Arrival Time: Oct 3, 2016 17:21:43.275218512 EDT
- [Time shift for this packet: 0.000000000 seconds]
- Epoch Time: 1475529703.275218512 seconds
- [Time delta from previous captured frame: 0.432126441 seconds]
- [Time delta from previous displayed frame: 0.432126441 seconds]
- [Time since reference or first frame: 12760.733404443 seconds]
- Frame Number: 31301
- Frame Length: 42 bytes (336 bits)
- Capture Length: 42 bytes (336 bits)
- [Frame is marked: False]
- [Frame is ignored: False]
- [Protocols in frame: eth:arp]
- [Coloring Rule Name: ARP]
- [Coloring Rule String: arp]
- Ethernet II, Src: Actionte_20:7a:f7 (00:7f:28:20:7a:f7), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
 - Destination: Broadcast (ff:ff:ff:ff:ff:ff)
 - Source: Actionte_20:7a:f7 (00:7f:28:20:7a:f7)
 - Type: ARP (0x0806)
- Address Resolution Protocol (request)
 - Hardware type: Ethernet (1)
 - Protocol type: IP (0x0800)
 - Hardware size: 6
 - Protocol size: 4

```
0000 ff ff ff ff ff 00 7f 28 20 7a f7 08 06 00 01
0010 08 00 06 04 00 01 00 7f 28 20 7a f7 c0 a8 01 01
0020 00 00 00 00 00 c0 a8 01 0b
```

And here are some of the details of a TCP packet:



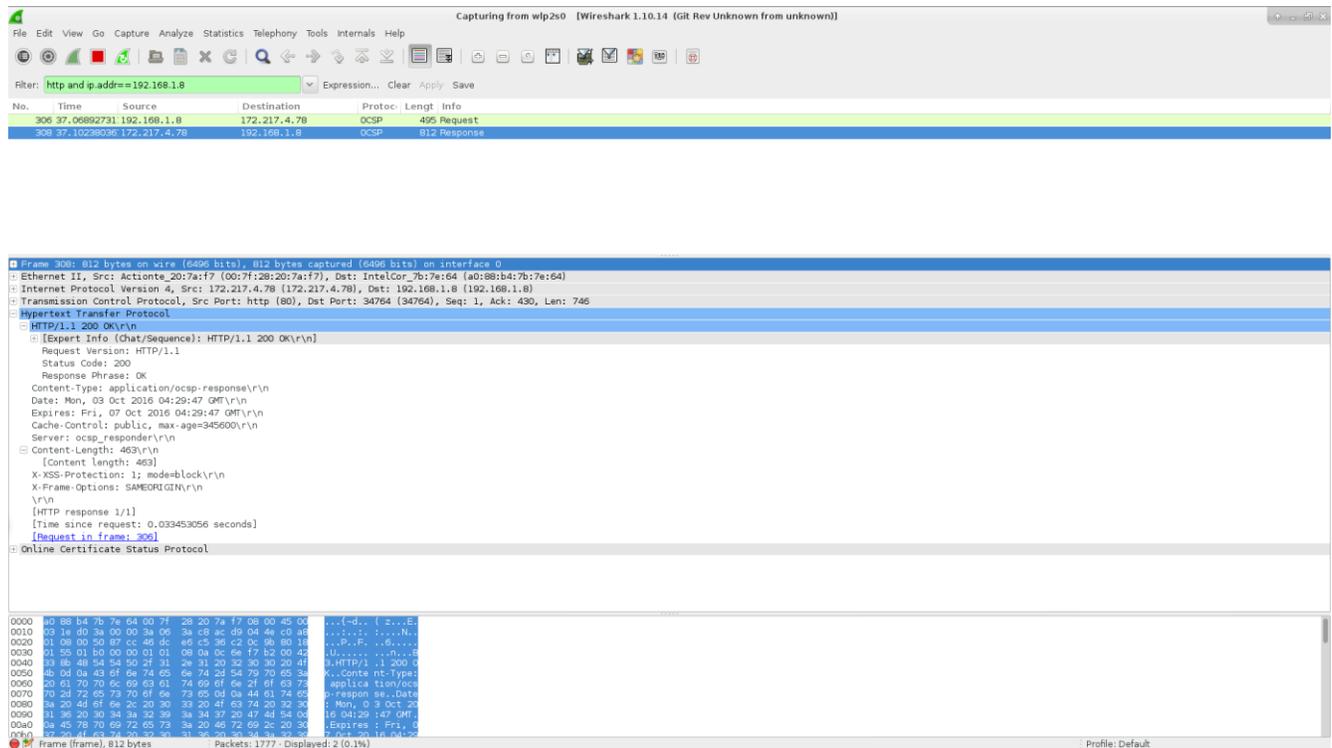
The information in the packets differs. Obviously the protocol is labeled differently, but also things like frame length, payload and the TCP packet gives information about what version it's using.

View Packet Data with the Individual Packet Bytes Window

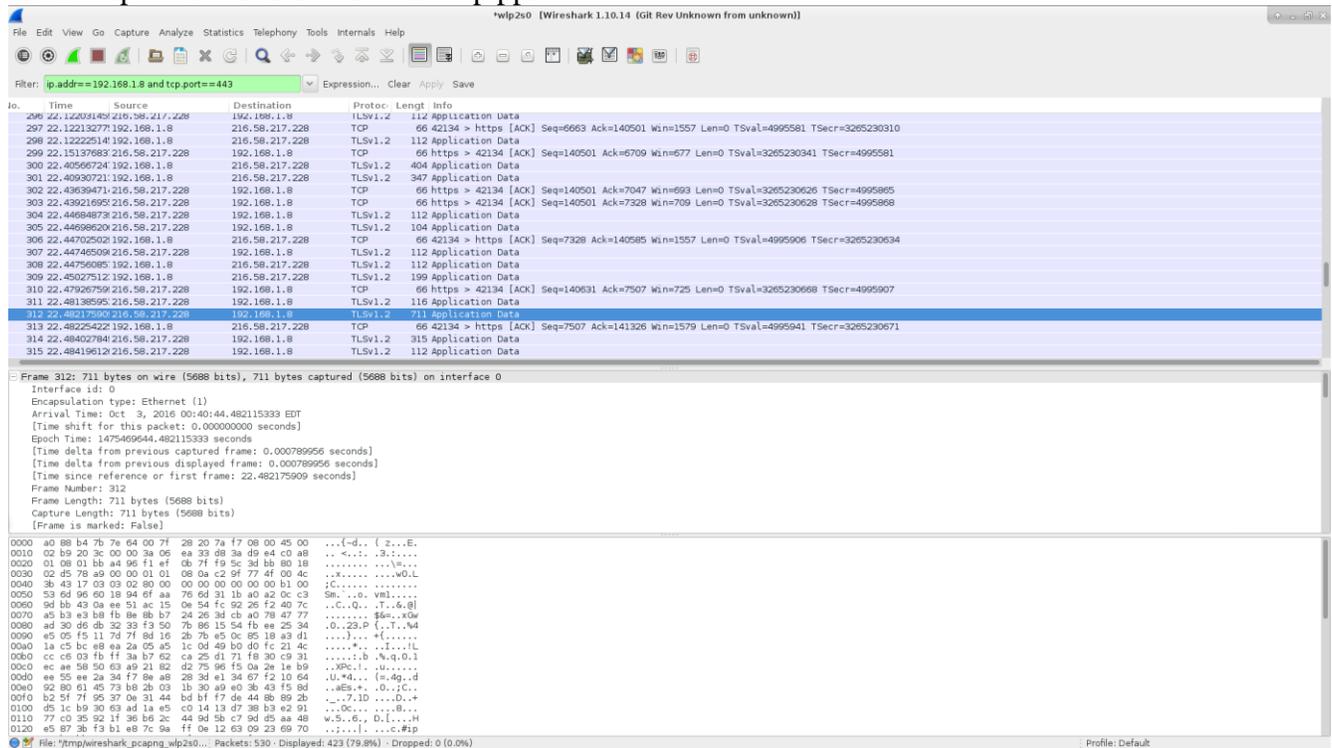
Using the above examples, the Packet Bytes window is the bottom frame. The data held in the packet bytes window is different because both packets are different. The same as opening a file in a hex editor, the hexadecimal representation is on the left side of the frame, and its ASCII representation is on the right side of the frame. There is a period (".") for hexadecimal information that does not have ASCII characters associated with it.

Browse The Internet

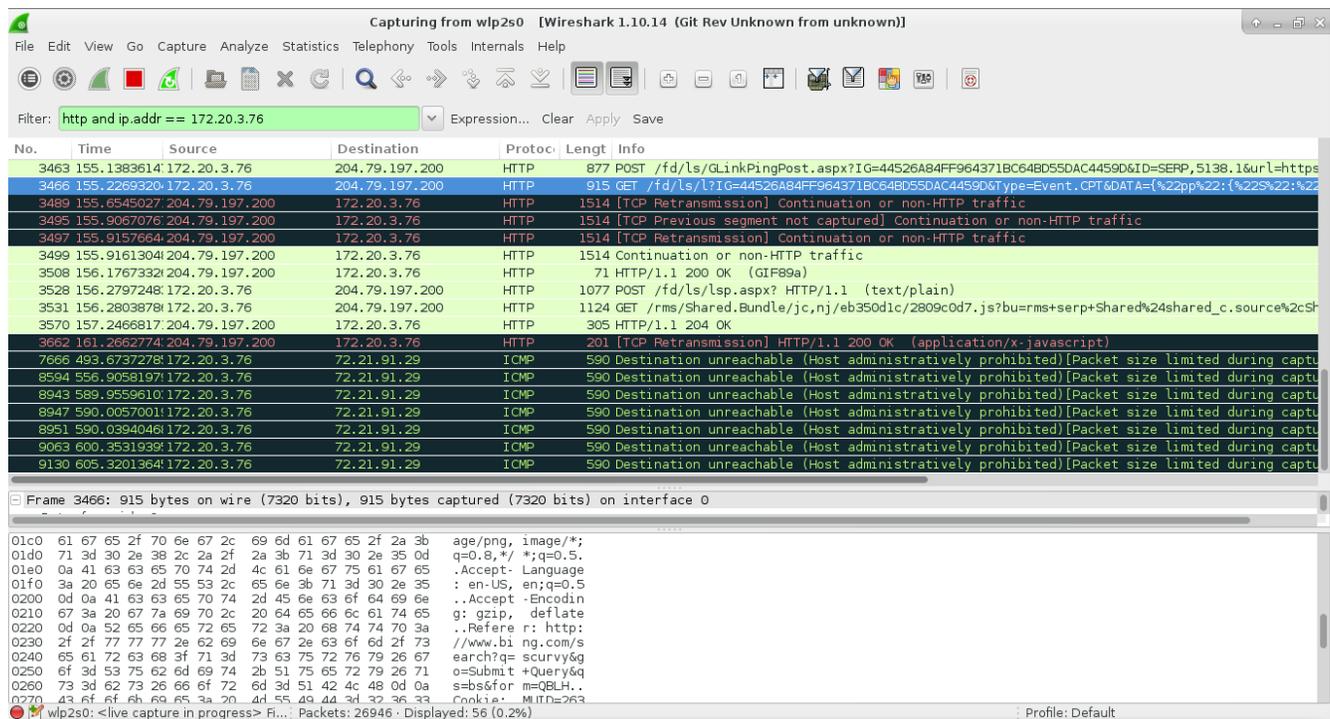
Filter on ip.addr==192.168.1.8 and http., visit <http://www.google.com> and perform a search on scurvy



When this action was performed, I found that Google redirected to https – port 443. I then changed my filter to ip.addr==192.168.1.8 and tcp.pprt==443:



After inspecting the packets I found that my search on scury was in fact secure using the SSL connection. I then tried the same experiment with the Microsoft Bing search engine:



In this case I was able to view my search within the packet because it was going over a non-encrypted connection. I was also able to view data from the Wikipedia page (first link) as it too was non-encrypted.

Analyze Wireshark Data

- a) How many UDP packets did Wireshark capture: 704

Wireshark: Protocol Hierarchy Statistics

Display filter: none

Protocol	% Packets	Packets	% Bytes	Bytes	Mbit/s	End Packets	End Bytes	End I
[-] Frame	100.00 %	8388	100.00 %	13866371	0.296	0	0	
[-] Ethernet	100.00 %	8388	100.00 %	13866371	0.296	0	0	
[-] Internet Protocol Version 6	0.56 %	47	0.03 %	4676	0.000	0	0	
Internet Control Message Protocol v6	0.48 %	40	0.03 %	3860	0.000	40	3860	
[-] User Datagram Protocol	0.08 %	7	0.01 %	816	0.000	0	0	
Domain Name Service	0.05 %	4	0.00 %	486	0.000	4	486	
DHCPv6	0.04 %	3	0.00 %	330	0.000	3	330	
[-] Internet Protocol Version 4	96.84 %	8123	99.90 %	13852063	0.295	0	0	
Internet Group Management Protocol	0.17 %	14	0.01 %	744	0.000	14	744	
[-] User Datagram Protocol	8.39 %	704	0.64 %	88489	0.002	0	0	
Domain Name Service	8.18 %	686	0.60 %	83412	0.002	686	83412	
Hypertext Transfer Protocol	0.14 %	12	0.03 %	4533	0.000	12	4533	
Network Time Protocol	0.05 %	4	0.00 %	360	0.000	4	360	
NetBIOS Name Service	0.02 %	2	0.00 %	184	0.000	2	184	
[+] Transmission Control Protocol	87.57 %	7345	99.16 %	13749821	0.293	4293	3818408	
[+] Internet Control Message Protocol	0.72 %	60	0.09 %	13009	0.000	58	11829	
Address Resolution Protocol	2.53 %	212	0.06 %	8904	0.000	212	8904	
[+] Logical-Link Control	0.05 %	4	0.00 %	470	0.000	0	0	
802.1X Authentication	0.02 %	2	0.00 %	258	0.000	2	258	

Help Close

b) what was the average IP Packet size: 1653.120 bytes

Wireshark: Summary

File
 Name: /tmp/wireshark_pcapng_wlp2s0_20161003011534_ZIL27g
 Length: 14148490 bytes
 Format: Wireshark/... - pcapng
 Encapsulation: Ethernet

Time
 First packet: 2016-10-03 01:15:34
 Last packet: 2016-10-03 01:21:49
 Elapsed: 00:06:15

Capture
 OS: Linux 3.10.0-327.36.1.el7.x86_64
 Capture application: Dumpcap 1.10.14 (Git Rev Unknown from unknown)

Capture file comments:

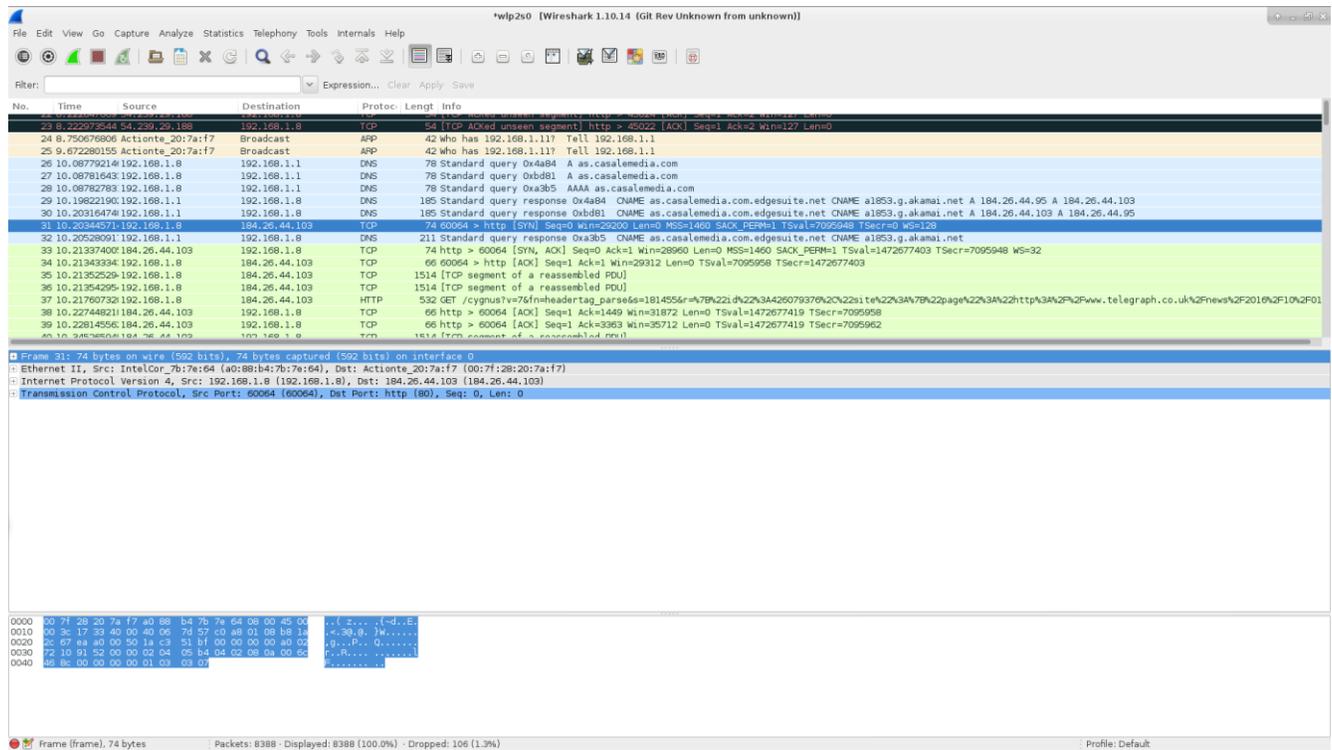
Interface	Dropped Packets	Capture Filter	Link type	Packet size limit
wlp2s0	unknown	none	Ethernet	262144 bytes

Display
 Display filter: none
 Ignored packets: 0 (0.000%)

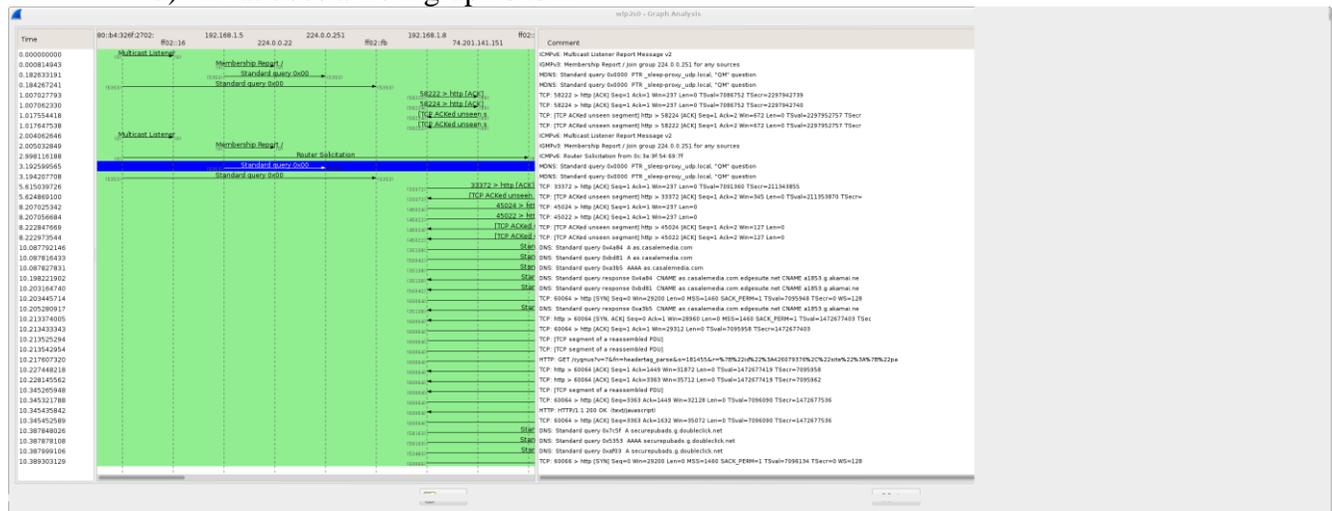
Traffic	Captured	Displayed	Displayed %	Marked	Marked %
Packets	8388	8388	100.000%	0	0.000%
Between first and last packet	375.319 sec				
Avg. packets/sec	22.349				
Avg. packet size	1653.120 bytes				
Bytes	13866371	13866371	100.000%	0	0.000%
Avg. bytes/sec	36945.528				
Avg. MBit/sec	0.296				

Help Cancel OK

c) how many packets did Wireshark drop:106

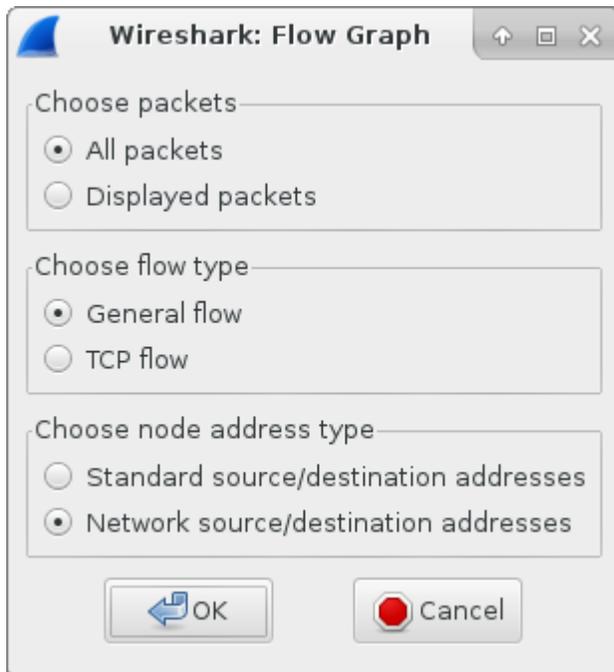


d) what does a flow graph show:



A flow graph shows the data flow of a connection. By scrolling to the right, I can see things like retransmits or drops.

e) list the flow graph options



Analysis Questions

1. How can security professionals use Wireshark?

By using Wireshark, you can view the raw data of a packet for file signatures we discussed in our hex lab, or spot instructions for execution of malicious code. Another way to use Wireshark is to discover DoS attacks or ssh brute force attacks. By capturing an overwhelming number of a certain type of incoming packets you can take steps to block the source IP address. If I saw a large number of packets going to a remote destination IP address, I would investigate to make sure corporate data was not being removed from an unauthorized source.

In addition, I could monitor incoming and outgoing traffic based on protocol and create rules. For SMTP as an example, I could monitor for attachments of a certain size or content.

2. List three ways attackers can use Wireshark.

- 1) To map the network
- 2) Discover protocols that are in use
- 3) Find clear text communications for username and password discovery

3. Does Wireshark capture all the traffic on the Internet? If so, explain why. If not, which traffic does it capture?

No, Wireshark captures traffic that passes its network interface either directly or on a hub, as well as broadcast and multipath traffic (Weadock, 2009).

4. Write Wireshark filters to:

- | | |
|--|------------------------------|
| a) View all traffic for 10.10.10.2. | Filter = ip.addr==10.10.10.2 |
| b) View icmp traffic from any address. | Filter = icmp |

5. Has this lab changed your perspective on your privacy while browsing the Internet? Describe why or why not.

It hasn't changed my perspective of privacy on the Internet; because of my job, I've been exposed to packet sniffers and understand their capabilities. That said, this lab gave me an opportunity to dive in a lot deeper than I had previously and look at some of the features that Wireshark has. So, basically I have a more, well rounded understanding of packet sniffing and filtering than I had previously.

Conclusion

Through the exercises, I learned about the capabilities of the Wireshark packet sniffer. I was able to capture packet data and inspect individual packets. I gained a better understanding about the contents of a packet, and how I can gather useful information from using the Wireshark sniffer and putting what I learned into the context of how a network is used and what kinds of data I'm expecting to see.

References

Adrian Hannah (2011 Nov. 14) Packet Sniffing Basics. Retrieved from <http://www.linuxjournal.com/content/packet-sniffing-basics>

Glenn Weadock (2009 Sept 30). Wireshark and Promiscuous Mode - Why you may not be seeing all the traffic you think you should. Retrieved from <http://www.networkworld.com/article/2231903/microsoft-subnet/wireshark-and-promiscuous-mode.html>

Paessler (1996) "Packet Sniffing; Packet Sniffing With PRTG Network Monitor" Retrieved from https://www.paessler.com/packet_sniffing

Craig Hunt (1992). TCP/IP Network Administration. Sebastopol, CA: O'Reilly & Associates

Douglas E. Comer (1997). Computer Networks and Internets with Internet Applications; Third Edition. Upper Saddle River, New Jersey. Prentice Hall

OpenBSD Documentation (2016 Sept 1), PF Users Guide, Retrieved from <https://www.openbsd.org/faq/pf/filter.html>

W. Richard Stevens (1990). UNIX Network Programming. Englewood Cliffs, NJ. Prentice Hall

Sahin Erbay 2016, July 26) Internet Protocol Suite (TCP/IP and OSI Models); Retrieved from <http://sahinerbay.com/2016/02/26/internet-protocol-suite-tcpip-and-osi-models/>

Wireshark (2008 April 12) Duplicate Packets. Retrieved from <https://wiki.wireshark.org/DuplicatePackets>

Wireshark (2014). Wireshark User's Guide. Retrieved from https://www.wireshark.org/docs/wsug_html_chunked/