Lab: Pfsense and Wireshark

Rich Macfarlane

Aim:

The aim of this lab is to configure a perimeter network with a PfSense firewall, and explore the Wireshark network packet analyser for network traffic investigation.

Time to Complete:

2-4 hours

Activities:

- Complete Labs

Learning activities:

At the end of this lab, you should understand:

- How to configure a Pfsense firewall, and set up the firewalling to allow traffic needed for network analysis/investigation.
- How to use Wireshark to capture and interpret network packet traces, analysing protocols such as the HTTP, Telnet and SSH , by searching and filtering, and manipulate and save pcap traces.

Reflective statements (end-of-exercise):

- How can a monitoring and traffic analysis tool such as Wireshark be used in a network security context?
- How might Wireshark statistics, filtering and pcap manipulation be used in a network analysis/investigation context.

References:

- Previous lab notes as reference for configuration if needed
- Pfsense online documentation: **https://doc.pfsense.org/index.php/Main_Page**
- The Wireshark User Guide can be found at: **http://www.wireshark.org/docs/wsug_html_chunked**

Setting up your VMs and Network

Figure 1 outlines the setup of the lab for each group/student. We will assign network address to the 3 interfaces of the firewall and associated interfaces of the hosts so we can route traffic between hosts. NAT is automatic in PFSense as previsouly explored, and will allow the DMZ and private networks to connect to the public network. A DNS nameserver will need to be configured on the Hosts.

Network Ranges:
Your networks will be: 10.10.x.0/24 and 10.10.x.0/24

First log into vCenter (vc2003.napier.ac.uk), and then select your network infrastructure. Use this to configure the network addresses for your firewall and hosts from the network ranges on Moodle.

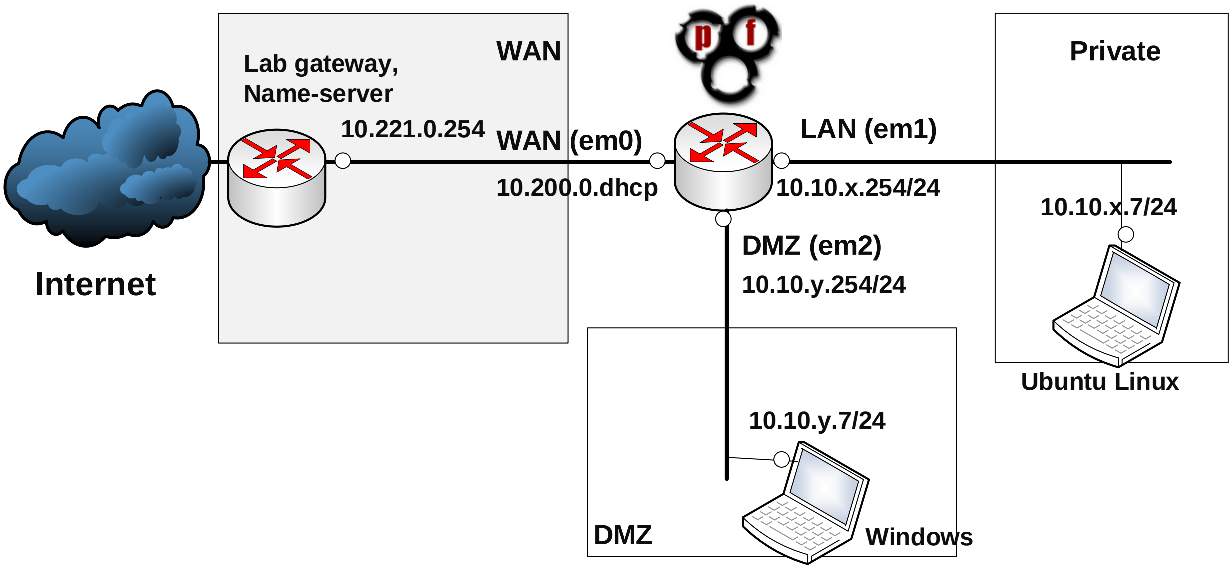


Figure 1: Individual Student Lab setup

User logins: Ubuntu (User: napier, Password: napier123),
Windows: (User: Administrator, Password: napier),
Pfsense (User: admin, Password: pfsense)
Kali (User: root, Password: toor)

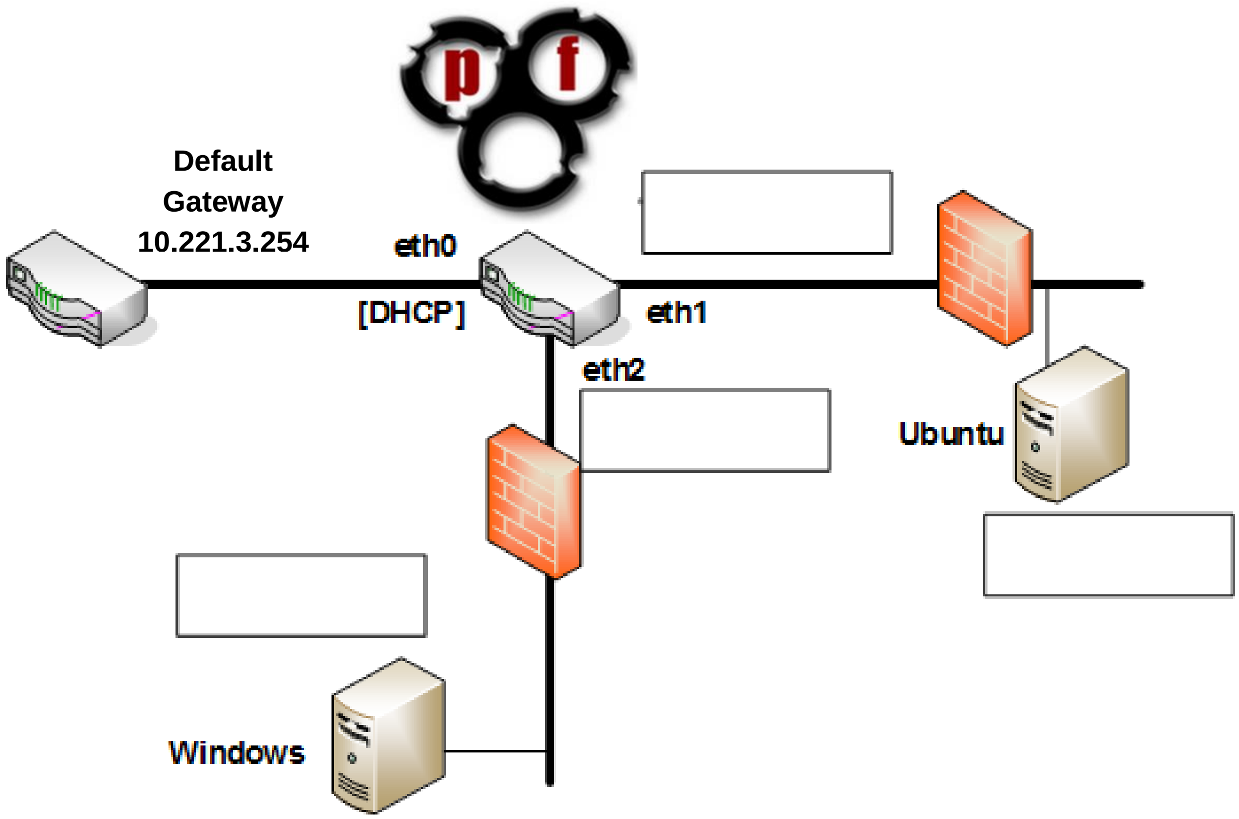


Figure 3: Each student or group’s network – complete the diagram with your own IP Addressing from table and refer back to this while doing the lab!!

Initial Configuration of the PFSense Firewall

Ref for configuring the PFSense firewall:
https://doc.pfsense.org/index.php/Main_Page

Create Interface Names/Configure Addressing

Create the interface names for the outside, inside and DMZ network interfaces, and then assign IP Addresses. DHCP for the outside, and static IP Addresses for the the Inside and DMZ, as previously. Check the IP Addresses are correct and test connectivity.

Test Connectivity

Q. Can you ping the WAN interface? Yes/No

Q. Can you ping the main gateway (10.221.3.254)? Yes/No

Test Connectivity

Q. Can you ping the LAN interface? Yes/No

Configure Host Networking

Now we will configure the hosts to sit on the Private and DMZ zones.

Setup the Linux host with an IP Address of 10.10.x.7/24 and a default gateway of your firewall LAN interface (10.10.x.254/24).

```
sudo ip addr add 10.10.x.7/24 dev ethx
sudo ip route add default via 10.10.x.254
```

Setup the DNS name server on the Linux host by editing the /etc/resolv.config and adding a nameserver:

```
sudo nano /etc/resolv.conf
```

Add the nameserver `10.221.3.254`

On the Windows server modify the static address on the interface with:

IP: 10.10.y.7
Subnet mask: 255.255.255.0
Gateway: 10.10.y.254
DNS: 10.221.3.254 – The lab default gateway should have a DNS server running

Test Connectivity

Using the **ping** command check the local interface are up and running and then connectivity from one system to another, by pinging each interface, starting with the local machine interface and working to the other hosts one hop at a time.

Firewall Remote Administration

Now we configure the firewall, via the webConfigurator web service running on the firewall. From Ubuntu, connect to the firewall using a web browser. Login with the pfsense credentials

Private LAN and DMZ Network Connectivity

Check your configuration against your network diagram, using **Status>Interfaces**
Go back to the Dashboard **Status > Dashboard**

LAN and DMZ Network Connectivity

Send each of the 10.10.x.254 and 10.10.x.7 interfaces some ICMP packets.

Q. Can you ping them? Yes/No

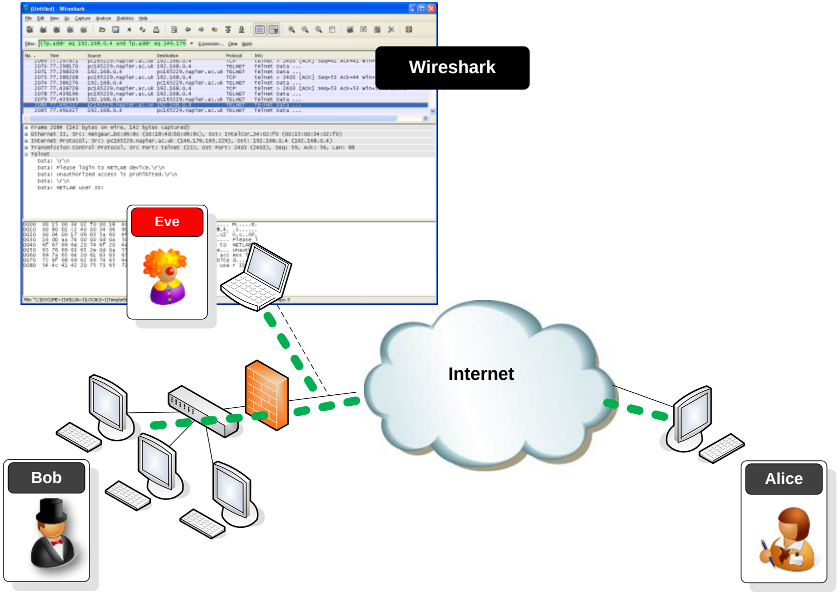
Firewall Rules

From the firewall web app, review the current rules applied to the 3 firewall interfaces using
Firewall>Rules (and then the appropriate interface tabs)
Create rules to allow DMZ to the Inside Private network for ALL traffic
Create rules to allow DMZ to the ANY network for ICMP, WEB, and DNS traffic
Also check the private and bogon checkboxes are unchecked in Interfaces>WAN

Wireshark - Capturing and Analysing Traffic

Packet Capture (Packet Sniffing)

A packet sniffer is an application which can capture and analyse network traffic which is passing through a system's Network Interface Card (NIC). The sniffer sets the card to promiscuous mode which means all traffic is read, whether it is addressed to that machine or not. The figure below shows an attacker sniffing packets from the network, and the Wireshark packet sniffer/analyser.


(Untitled) - Wireshark
File Edit View Go Capture Analyze Statistics Help
Filter: Expression... Clear Apply
Wireshark
Eve
Internet
Bob
Alice

```
Filter: (ip.addr eq 192.168.0.4 and ip.addr eq 146.176

No. .  Time        Source                 Destination            Protocol Info
2070 77.298170  pc165229.napier.ac.uk  192.168.0.4            TELNET   Telnet Data ...
2071 77.298329  192.168.0.4            pc165229.napier.ac.uk  TELNET   Telnet Data ...
2073 77.386208  pc165229.napier.ac.uk  192.168.0.4            TCP      telnet > 2403 [ACK] Seq=53 Ack=44 Win=
2074 77.386276  192.168.0.4            pc165229.napier.ac.uk  TELNET   Telnet Data ...
2077 77.438728  pc165229.napier.ac.uk  192.168.0.4            TCP      telnet > 2403 [ACK] Seq=53 Ack=53 Win=
2078 77.439196  pc165229.napier.ac.uk  192.168.0.4            TELNET   Telnet Data ...
2079 77.439345  192.168.0.4            pc165229.napier.ac.uk  TELNET   Telnet Data ...
2085 77.491427  192.168.0.4            pc165229.napier.ac.uk  TELNET   Telnet Data ...

Frame 2084 (142 bytes on wire, 142 bytes captured)
Ethernet II, Src: Netgear_b0:d6:8c (00:18:4d:b0:d6:8c), Dst: IntelCor_34:02:f0 (00:15:00:34:02:f0)
Internet Protocol, Src: pc165229.napier.ac.uk (146.176.165.229), Dst: 192.168.0.4 (192.168.0.4)
Transmission Control Protocol, Src Port: telnet (23), Dst Port: 2403 (2403), Seq: 59, Ack: 56, Len: 88
Telnet
    Data: \r\n
    Data: Please login to NETLAB device.\r\n
    Data: Unauthorized access is prohibited.\r\n
    Data: \r\n
    Data: NETLAB user ID:
```

Packet Analysis

Wireshark is an open source cross-platform packet capture and analysis tool, with versions for Windows and Linux. The GUI window gives a detailed breakdown of the network protocol stack for each packet, colorising packet details based on protocol, as well as having functionality to filter and search the traffic, and pick out TCP streams. Wireshark can also save packet data to files for offline analysis and export/import packet captures to/from other tools. Statistics can also be generated for packet capture files.

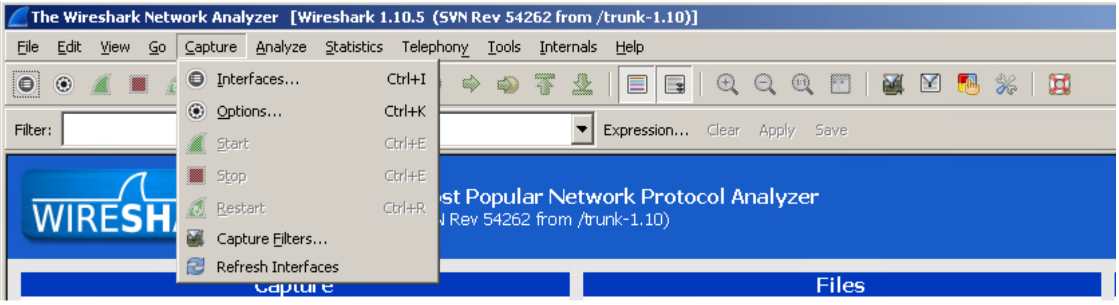
Wireshark can be used for **network troubleshooting**, to **investigate security issues**, and to **analyse and understand network protocols**. The packet sniffer can exploit information passed in plaintext, i.e. not encrypted. Examples of **protocols** which pass information in plaintext are **Telnet, FTP, SNMP, POP, and HTTP**.

Wireshark is a GUI based network capture tool. There is a command line based version of the packet capture utility, called **TShark**. TShark provides many of the same features as its big brother, but is console-based. It can be a good alternative if only command line access is available, and also uses less resources as it has no GUI to generate.

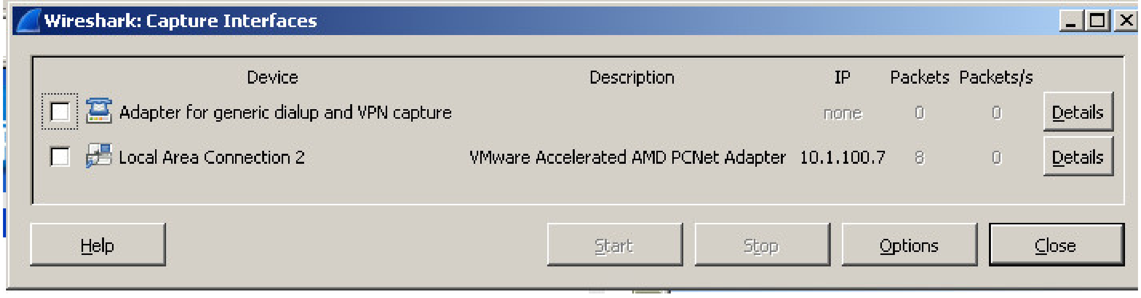
The Wireshark User Guide can be found at:
http://www.wireshark.org/docs/wsug_html_chunked/

Capturing Network Traffic

From your Windows Server VM, start the Wireshark application. When Wireshark is first run, a default, or blank window is shown. To list the available network interfaces, select the **Capture->Interfaces** menu option (CTRL+I).


The Wireshark Network Analyzer [Wireshark 1.10.5 (SVN Rev 54262 from /trunk-1.10)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Interfaces... Ctrl+I
Options... Ctrl+K
Start Ctrl+E
Stop Ctrl+E
Restart Ctrl+R
Capture Filters...
Refresh Interfaces
Filter:
Expression... Clear Apply Save
WIRESH
st Popular Network Protocol Analyzer
Rev 54262 from /trunk-1.10)
Capture
Files

Wireshark should display an Interfaces popup window such as the one shown below.



Device	Description	IP	Packets	Packets/s	
□ Adapter for generic dialup and VPN capture		none	0	0	Details
□ Local Area Connection 2	VMware Accelerated AMD PCNet Adapter	10.1.100.7	8	0	Details

Ping the nearest Windows machine's default gateway, and watch the Packets column.

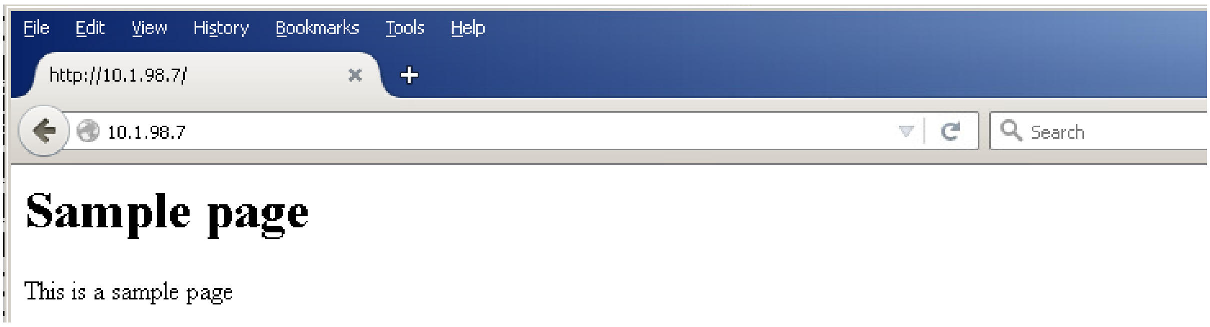
Questions

Q. Which Interface is connected to a local network (Ethernet)?

Q. How many packets have passed through the interface?

To capture network traffic click the **Start** button for the network interface you want to capture traffic on. Windows can have a long list of virtual interfaces, before the Ethernet Network Interface Card (NIC).

From the Windows VM, generate some network traffic with a Web Browser, such as Firefox, and access the web server running on your Ubuntu VM.

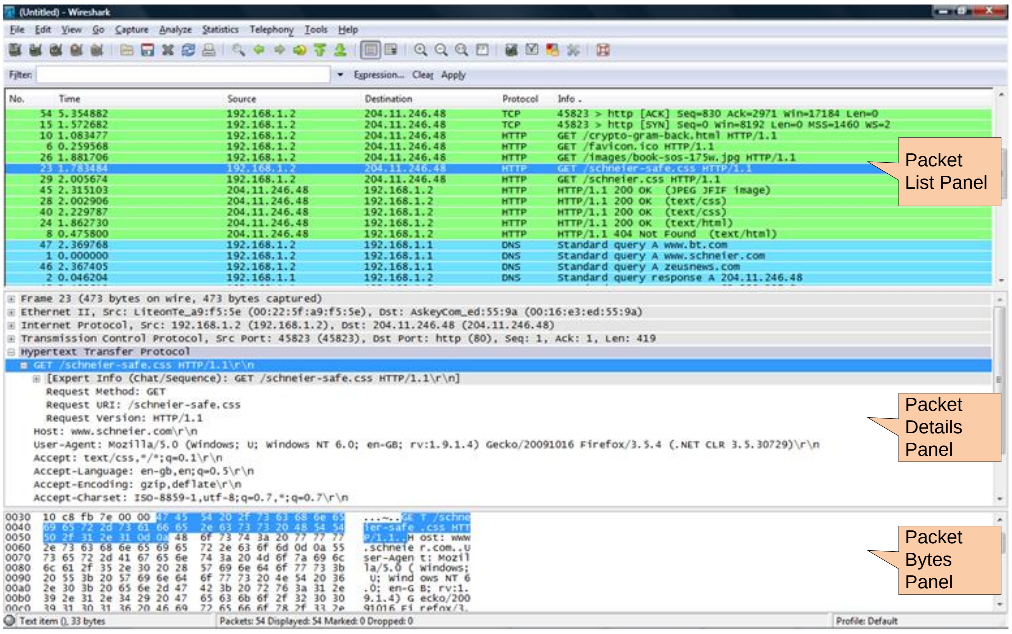


Sample page

This is a sample page

Wireshark Interface

Your Wireshark window should show the packets, and now look something like the following:


(Untitled) - Wireshark
File Edit View Go Capture Analyze Statistics Telephony Tools Help
Filter:
Expression... Clear Apply
Packet List Panel
Packet Details Panel
Packet Bytes Panel

```
No.     Time          Source          Destination     Protocol  Info .
     54 5.354882      192.168.1.2     204.11.246.48   TCP       45823 > http [ACK] Seq=830 Ack=2971 Win=17184 Len=0
     15 1.572682      192.168.1.2     204.11.246.48   TCP       45823 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=2
     10 1.083477      192.168.1.2     204.11.246.48   HTTP      GET /crypto-gram-back.html HTTP/1.1
      6 0.259568      192.168.1.2     204.11.246.48   HTTP      GET /favicon.ico HTTP/1.1
     26 1.881706      192.168.1.2     204.11.246.48   HTTP      GET /images/book-sos-175w.jpg HTTP/1.1
     23 1.783484      192.168.1.2     204.11.246.48   HTTP      GET /schneier-safe.css HTTP/1.1
     29 2.005674      192.168.1.2     204.11.246.48   HTTP      GET /schneier.css HTTP/1.1
     45 2.315103      204.11.246.48   192.168.1.2     HTTP      HTTP/1.1 200 OK  (JPEG JFIF image)
     28 2.002906      204.11.246.48   192.168.1.2     HTTP      HTTP/1.1 200 OK  (text/css)
     40 2.229787      204.11.246.48   192.168.1.2     HTTP      HTTP/1.1 200 OK  (text/css)
     24 1.862730      204.11.246.48   192.168.1.2     HTTP      HTTP/1.1 200 OK  (text/html)
      8 0.475800      204.11.246.48   192.168.1.2     HTTP      HTTP/1.1 404 Not Found  (text/html)
     47 2.369768      192.168.1.2     192.168.1.1     DNS       Standard query A www.bt.com
      1 0.000000      192.168.1.2     192.168.1.1     DNS       Standard query A www.schneier.com
     46 2.367405      192.168.1.2     192.168.1.1     DNS       Standard query A zeusnews.com
      2 0.046204      192.168.1.1     192.168.1.2     DNS       Standard query response A 204.11.246.48
```

```
⊞ Frame 23 (473 bytes on wire, 473 bytes captured)
⊞ Ethernet II, Src: LiteonTe_a9:f5:5e (00:22:5f:a9:f5:5e), Dst: AskeyCom_ed:55:9a (00:16:e3:ed:55:9a)
⊞ Internet Protocol, Src: 192.168.1.2 (192.168.1.2), Dst: 204.11.246.48 (204.11.246.48)
⊞ Transmission Control Protocol, Src Port: 45823 (45823), Dst Port: http (80), Seq: 1, Ack: 1, Len: 419
⊟ Hypertext Transfer Protocol
  ⊟ GET /schneier-safe.css HTTP/1.1\r\n
    ⊞ [Expert Info (Chat/Sequence): GET /schneier-safe.css HTTP/1.1\r\n]
      Request Method: GET
      Request URI: /schneier-safe.css
      Request Version: HTTP/1.1
    Host: www.schneier.com\r\n
    User-Agent: Mozilla/5.0 (Windows; U; Windows NT 6.0; en-GB; rv:1.9.1.4) Gecko/20091016 Firefox/3.5.4 (.NET CLR 3.5.30729)\r\n
    Accept: text/css,*/*;q=0.1\r\n
    Accept-Language: en-gb,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
```

```
0030  10 c8 fb 7e 00 00 47 45  54 20 2f 73 63 68 6e 65   ...~..GE T /schne
0040  69 65 72 2d 73 61 66 65  2e 63 73 73 20 48 54 54   ier-safe .css HTT
0050  50 2f 31 2e 31 0d 0a 48  6f 73 74 3a 20 77 77 77   P/1.1..H ost: www
0060  2e 73 63 68 6e 65 69 65  72 2e 63 6f 6d 0d 0a 55   .schneie r.com..U
0070  73 65 72 2d 41 67 65 6e  74 3a 20 4d 6f 7a 69 6c   ser-Agen t: Mozil
0080  6c 61 2f 35 2e 30 20 28  57 69 6e 64 6f 77 73 3b   la/5.0 ( Windows;
0090  20 55 3b 20 57 69 6e 64  6f 77 73 20 4e 54 20 36    U; Wind ows NT 6
00a0  2e 30 3b 20 65 6e 2d 47  42 3b 20 72 76 3a 31 2e   .0; en-G B; rv:1.
00b0  39 2e 31 2e 34 29 20 47  65 63 6b 6f 2f 32 30 30   9.1.4) G ecko/200
00c0  39 31 30 31 36 20 46 69  72 65 66 6f 78 2f 33 2e   91016 Fi refox/3.
```

```
Text item (), 33 bytes    Packets: 54 Displayed: 54 Marked: 0 Dropped: 0    Profile: Default
```

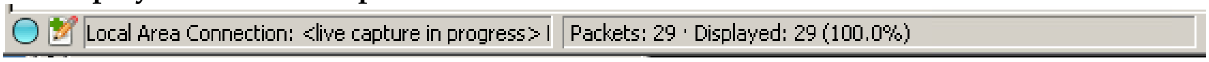
To stop the capture, select the **Capture->Stop** menu option, Ctrl+E, or the Stop toolbar button. What you have created is a Packet Capture or ***'pcap'***, which you can now view and analyse using the Wireshark interface, or save to disk to analyse later.

The capture is split into 3 parts:

1. **Packet List Panel –** this is a list of packets in the current capture. It colours the packets based on the protocol type.
   For each packet the columns include:
   - **No** – order the pkts were received by Wireshark
   - **Time** – UNIX Time – this can be configured to a timezone and used to timeline incidents!
   - **Src and Dest IP Address**
   - **Protocol – DNS, ICMP, HTTP ARP etc**
   - **Information** – about the specific pkt – this is interpreted by Wireshark!

   When a packet is selected, the details are shown in the two panels below:
2. **Packet Details Panel –** this shows the details of the selected packet. It shows the different protocols making up the layers of data for this packet. Layers include Frame, Ethernet, IP, TCP/UDP/ICMP, and application protocols such as HTTP.
3. **Packet Bytes Panel –** shows the packet bytes in Hex and ASCII encodings.

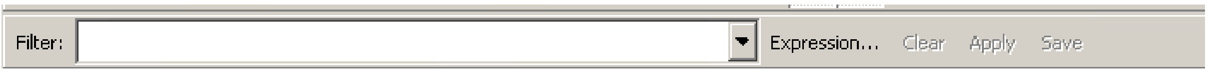
The status bar at the foot of the window shows **total number of packets in the capture**, and the **displayed** and **marked packets**!



```
Local Area Connection: <live capture in progress> |  Packets: 29 · Displayed: 29 (100.0%)
```

Wireshark Traffic Filtering

The Wireshark window also has the 'traffic **Filter field'** which can be used to enter a 'display filter', which then changes the traffic displayed to only show packets which match the filter.



Some useful Wireshark **display filters** can be found at:
http://wiki.wireshark.org/DisplayFilters
https://www.wireshark.org/docs/dfref/

The simplest filtering would be at the lower layers of the TCP/IP model, such as filtering out based on IP Address.

Try filtering out only the traffic to/from your Windows VM, by entering a filter on the IP Address into the traffic Filter field:

```
        ip.addr == 10.10.y.7
```

Questions

Q: How many packets are in the capture, and how many are being displayed?

Try filtering out for only outgoing and then incoming traffic from the Windows VM using the following:

```
        ip.src == 10.10.y.7
        ip.dst == 10.10.y.7
```

Questions

Q: Check how many packets are in the capture, and how many are being displayed?

Out: In:

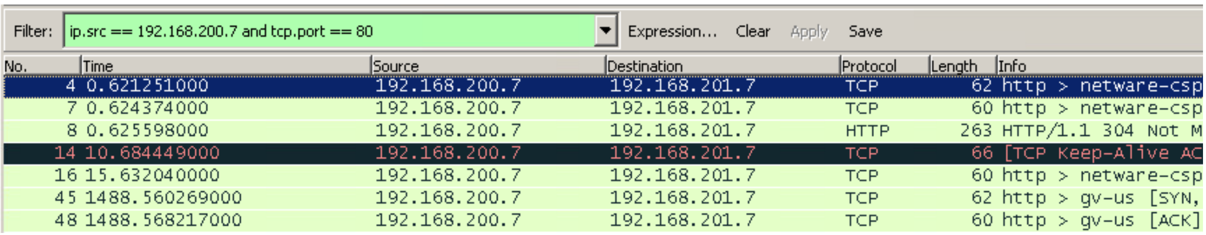
We can also build more complex filters which combine simpler filters. Try to create a filter to display **only packets coming back from a web server** on the private inside network. (tcp.port might be useful!)

Test by connecting to the Ubuntu web server a few times.

Questions

Q: What is the working filter?

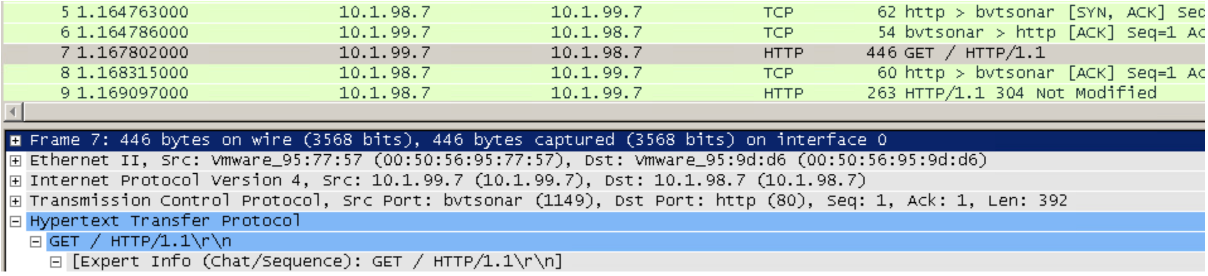
Something similar to the following should work: (using your IP addressing)



Filter: ip.src == 192.168.200.7 and tcp.port == 80

No.	Time	Source	Destination	Protocol	Length	Info
4	0.621251000	192.168.200.7	192.168.201.7	TCP	62	http > netware-csp
7	0.624374000	192.168.200.7	192.168.201.7	TCP	60	http > netware-csp
8	0.625598000	192.168.200.7	192.168.201.7	HTTP	263	HTTP/1.1 304 Not M
14	10.684449000	192.168.200.7	192.168.201.7	TCP	66	[TCP Keep-Alive AC
16	15.632040000	192.168.200.7	192.168.201.7	TCP	60	http > netware-csp
45	1488.560269000	192.168.200.7	192.168.201.7	TCP	62	http > gv-us [SYN,
48	1488.568217000	192.168.200.7	192.168.201.7	TCP	60	http > gv-us [ACK]

Search back through your capture, and find an **HTTP** packet containing an **HTTP GET** command. Click on the packet in the **Packet List Panel**. Then expand the HTTP layer in the **Packet Details Panel**, from the packet.



```
  5 1.164763000      10.1.98.7      10.1.99.7      TCP       62 http > bvtsonar [SYN, ACK] Sec
  6 1.164786000      10.1.99.7      10.1.98.7      TCP       54 bvtsonar > http [ACK] Seq=1 Ac
  7 1.167802000      10.1.99.7      10.1.98.7      HTTP     446 GET / HTTP/1.1
  8 1.168315000      10.1.98.7      10.1.99.7      TCP       60 http > bvtsonar [ACK] Seq=1 Ac
  9 1.169097000      10.1.98.7      10.1.99.7      HTTP     263 HTTP/1.1 304 Not Modified

⊞ Frame 7: 446 bytes on wire (3568 bits), 446 bytes captured (3568 bits) on interface 0
⊞ Ethernet II, Src: Vmware_95:77:57 (00:50:56:95:77:57), Dst: Vmware_95:9d:d6 (00:50:56:95:9d:d6)
⊞ Internet Protocol Version 4, Src: 10.1.99.7 (10.1.99.7), Dst: 10.1.98.7 (10.1.98.7)
⊞ Transmission Control Protocol, Src Port: bvtsonar (1149), Dst Port: http (80), Seq: 1, Ack: 1, Len: 392
⊟ Hypertext Transfer Protocol
  ⊟ GET / HTTP/1.1\r\n
    ⊟ [Expert Info (Chat/Sequence): GET / HTTP/1.1\r\n]
```

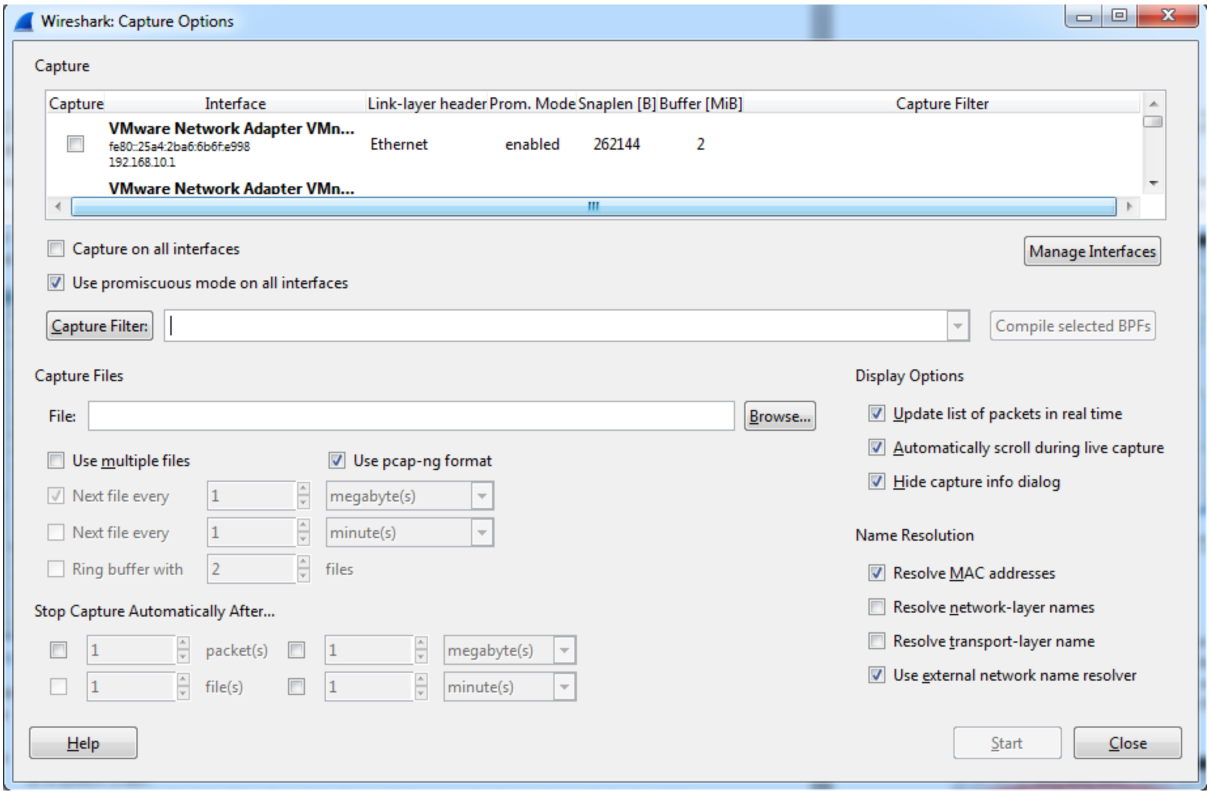
Questions

Q. From the **Packet Details Panel**, within the GET command, what is the value of the **Host param**?

Q. Can you see the **Hex** and **ASCII** representations of the packet in the **Packet Bytes Panel**?

Q. From the **Packet Bytes Panel**, what are the first 4 bytes of the Hex value of the **Host** parameter?

To select more detailed options when starting a capture, select the **Capture->Options** menu option, or **Ctrl+K**, or the Capture Options button on the toolbar (the wrench). This should show a window such as shown below.



Wireshark: Capture Options

Capture

Capture	Interface	Link-layer header	Prom. Mode	Snaplen [B]	Buffer [MiB]	Capture Filter
□	**VMware Network Adapter VMn...** fe80::25a4:2ba6:6b6f:e998 192.168.10.1	Ethernet	enabled	262144	2	
	VMware Network Adapter VMn...					

□ Capture on all interfaces

☑ Use promiscuous mode on all interfaces

Capture Filter:

Capture Files

File:

□ Use multiple files

☑ Use pcap-ng format

☑ Next file every 1 megabyte(s)

□ Next file every 1 minute(s)

□ Ring buffer with 2 files

Stop Capture Automatically After...

□ 1 packet(s) □ 1 megabyte(s)

□ 1 file(s) □ 1 minute(s)

Display Options

☑ Update list of packets in real time

☑ Automatically scroll during live capture

☑ Hide capture info dialog

Name Resolution

☑ Resolve MAC addresses

□ Resolve network-layer names

□ Resolve transport-layer name

☑ Use external network name resolver

Some of the more interesting options are:

- *Capture Options > Interface* **-** Again the important thing is to select the correct Network Interface to capture traffic through.
- *Capture Options > Capture File* **–** useful to save a file of the packet capture in real time, in case of a system crash.
- *Display Options > Update list of packets in real time* **–** A display option, which should be checked if you want to view the capture as it happens (typically switched off to capture straight to a file, for later analysis).
- *Name Resolution > MAC name resolution* – resolves the first 3 bytes of the MAC Address, the Organisation Unique Identifier (OUI), which represents the Manufacturer of the Card.
- *Name Resolution > Network name resolution* – does a DNS lookup for the IP Addresses captured, to display the network name. Set to off by default, so covert scans do not generate this DNS traffic, and tip off who's packets you are sniffing.

Questions

Q: Why might capture filters be useful/used as well as display filters?

Make sure the **MAC name resolution** is selected. Start the capture, and generate some Web traffic again, then stop the capture.

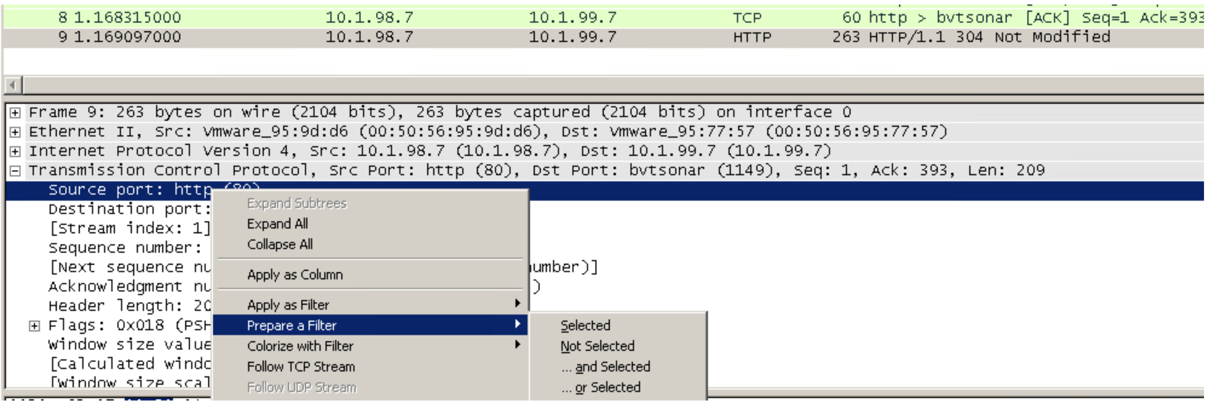
Questions

Search through your capture, and find an HTTP packet coming back from the server (TCP Source Port == 80). Expand the Ethernet layer in the **Packet Details Panel.**

Q: What is the OUI of the Network Interface Card (NIC)?

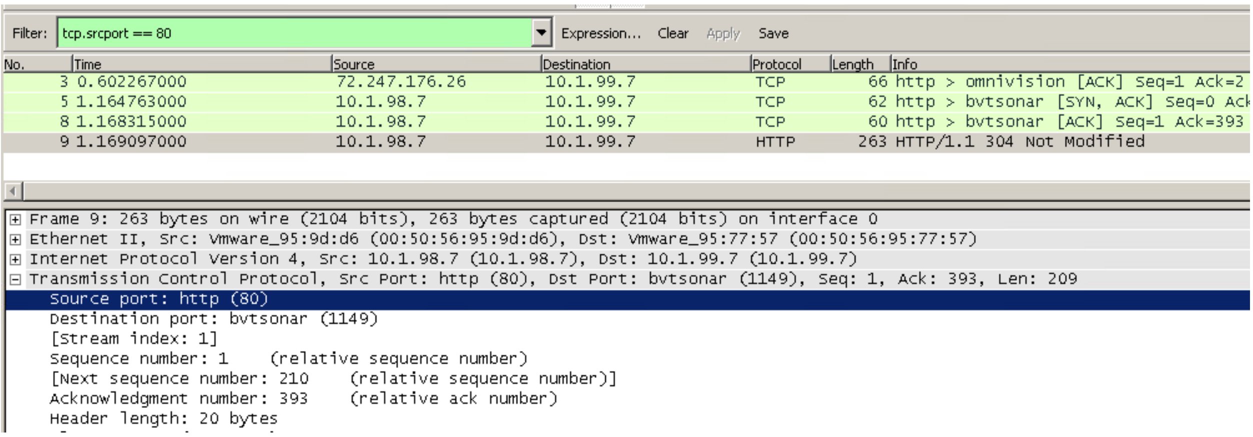
Q: What are the Hex values (shown the raw bytes panel) of the NICS Manufacturers OUIs?

Expand the TCP layer details in the **Packet Details Panel.** Right click on the **Source Port** field in the **Packet Details Panel**. Select **Prepare a Filter->Selected**.



```
8 1.168315000    10.1.98.7    10.1.99.7    TCP     60 http > bvtsonar [ACK] Seq=1 Ack=393
9 1.169097000    10.1.98.7    10.1.99.7    HTTP   263 HTTP/1.1 304 Not Modified

⊞ Frame 9: 263 bytes on wire (2104 bits), 263 bytes captured (2104 bits) on interface 0
⊞ Ethernet II, Src: Vmware_95:9d:d6 (00:50:56:95:9d:d6), Dst: Vmware_95:77:57 (00:50:56:95:77:57)
⊞ Internet Protocol Version 4, Src: 10.1.98.7 (10.1.98.7), Dst: 10.1.99.7 (10.1.99.7)
⊟ Transmission Control Protocol, Src Port: http (80), Dst Port: bvtsonar (1149), Seq: 1, Ack: 393, Len: 209
    Source port: http (80)
    Destination port:
    [Stream index: 1]
    Sequence number:
    [Next sequence nu                          umber)]
    Acknowledgment nu                          )
    Header length: 20
  ⊞ Flags: 0x018 (PSH
    Window size value
    [Calculated windo
    [Window size scal
```

Wireshark automatically generates a **Display Filter**. Click Apply and it is applied it to the capture. The filter is shown in the **Filter Bar**, below the button toolbar. Only packets captured with a Source Port of the value selected should be displayed. The window should be similar to that shown below. This same process can be performed on most fields within Wireshark, and can be used to include or exclude traffic.



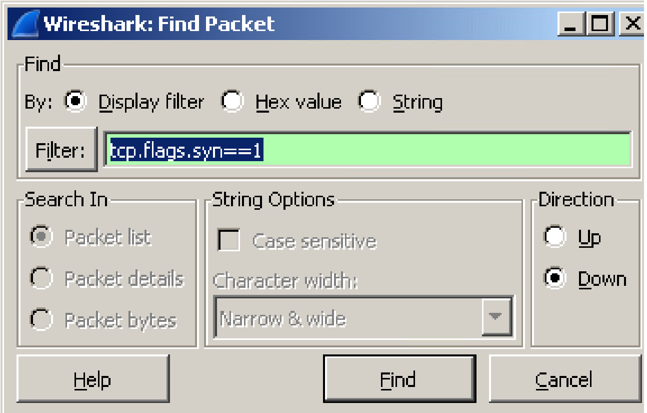
```
Filter: tcp.srcport == 80

No.  Time            Source          Destination   Protocol  Length  Info
  3 0.602267000     72.247.176.26   10.1.99.7     TCP          66 http > omnivision [ACK] Seq=1 Ack=2
  5 1.164763000     10.1.98.7       10.1.99.7     TCP          62 http > bvtsonar [SYN, ACK] Seq=0 Ack
  8 1.168315000     10.1.98.7       10.1.99.7     TCP          60 http > bvtsonar [ACK] Seq=1 Ack=393
  9 1.169097000     10.1.98.7       10.1.99.7     HTTP        263 HTTP/1.1 304 Not Modified

⊞ Frame 9: 263 bytes on wire (2104 bits), 263 bytes captured (2104 bits) on interface 0
⊞ Ethernet II, Src: Vmware_95:9d:d6 (00:50:56:95:9d:d6), Dst: Vmware_95:77:57 (00:50:56:95:77:57)
⊞ Internet Protocol Version 4, Src: 10.1.98.7 (10.1.98.7), Dst: 10.1.99.7 (10.1.99.7)
⊟ Transmission Control Protocol, Src Port: http (80), Dst Port: bvtsonar (1149), Seq: 1, Ack: 393, Len: 209
    Source port: http (80)
    Destination port: bvtsonar (1149)
    [Stream index: 1]
    Sequence number: 1    (relative sequence number)
    [Next sequence number: 210    (relative sequence number)]
    Acknowledgment number: 393    (relative ack number)
    Header length: 20 bytes
```

Analysing a TCP Session using Wireshark.

Start another traffic capture. Generate some Web traffic by going to **www.napier.ac.uk or another static page using HTTP rather than the encrypted HTTPS**, then stop the capture.

Scroll back to the top of the capture trace. Try to find the first SYN packet, sent from your PC to the Web Server. This signifies the start of a TCP 3-way handshake.

If you’re having trouble finding the first SYN packet, select the **Edit->Find Packet** menu option. Select the **Display Filter** radio button and enter a filter of **tcp.flags.** (at this point you should get a list of the flags to choose from). Choose the correct flag, **tcp.flags.syn** and add **== 1**.



Wireshark: Find Packet

Find

By: ● Display filter ○ Hex value ○ String

Filter: tcp.flags.syn==1

Search In: ● Packet list ○ Packet details ○ Packet bytes

String Options: ☐ Case sensitive

Character width: Narrow & wide

Direction: ○ Up ● Down

Hit the **Find** button, and the first SYN packet in the trace should be highlighted.
Note: Find Packet can also be used to search for a Hex signature, such as a malware signature, or to search for a string – such as a protocol command, or payload text - in the Packet Capture (**pcap**).

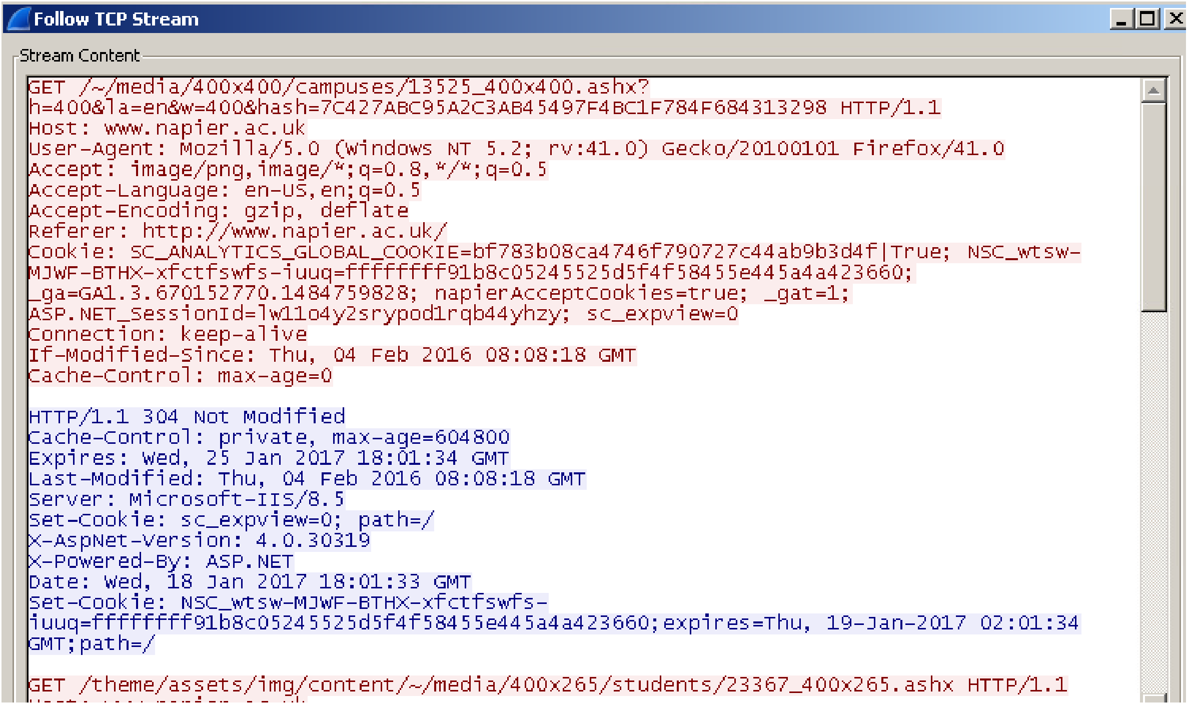
Questions

Q: Can you identify the rest of the TCP 3-way handshake easily? (if not read on)

A quick way to create a **Wireshark Display Filter** to isolate a TCP stream is to **right click** on a packet in the **Packet List Panel** and select **Follow TCP Stream.** This creates an automatic Display Filter which displays packets from that TCP session only.

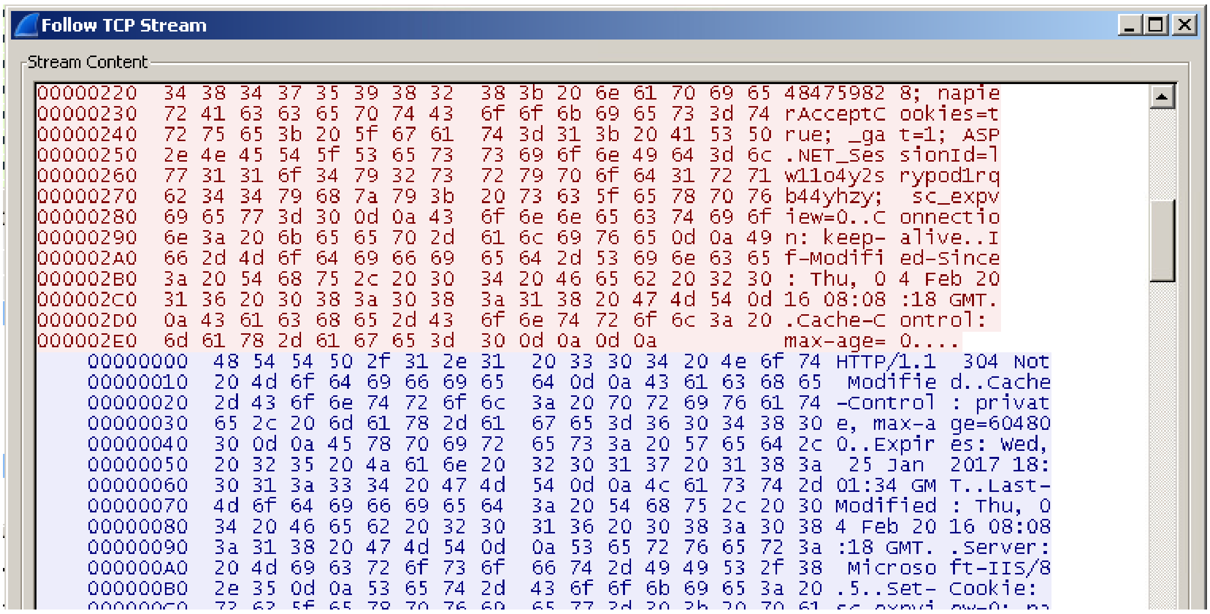
It also pops up a session display window, containing by default, an ASCII representation of the TCP session with the client packets in red and the server packets in blue.

The window should look something like shown below. This is very useful for viewing human readable protocol payloads, such as HTTP, SMTP, and FTP, or to check if the payloas is encrypted!



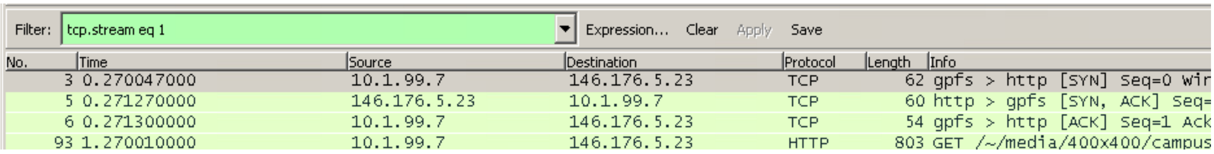
```
GET /~/media/400x400/campuses/13525_400x400.ashx?
h=400&la=en&w=400&hash=7C427ABC95A2C3AB45497F4BC1F784F684313298 HTTP/1.1
Host: www.napier.ac.uk
User-Agent: Mozilla/5.0 (Windows NT 5.2; rv:41.0) Gecko/20100101 Firefox/41.0
Accept: image/png,image/*;q=0.8,*/*;q=0.5
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://www.napier.ac.uk/
Cookie: SC_ANALYTICS_GLOBAL_COOKIE=bf783b08ca4746f790727c44ab9b3d4f|True; NSC_wtsw-
MJWF-BTHX-xfctfswfs-iuuq=ffffffff91b8c05245525d5f4f58455e445a4a423660;
_ga=GA1.3.670152770.1484759828; napierAcceptCookies=true; _gat=1;
ASP.NET_SessionId=lw11o4y2srypod1rqb44yhzy; sc_expview=0
Connection: keep-alive
If-Modified-Since: Thu, 04 Feb 2016 08:08:18 GMT
Cache-Control: max-age=0

HTTP/1.1 304 Not Modified
Cache-Control: private, max-age=604800
Expires: Wed, 25 Jan 2017 18:01:34 GMT
Last-Modified: Thu, 04 Feb 2016 08:08:18 GMT
Server: Microsoft-IIS/8.5
Set-Cookie: sc_expview=0; path=/
X-AspNet-Version: 4.0.30319
X-Powered-By: ASP.NET
Date: Wed, 18 Jan 2017 18:01:33 GMT
Set-Cookie: NSC_wtsw-MJWF-BTHX-xfctfswfs-
iuuq=ffffffff91b8c05245525d5f4f58455e445a4a423660;expires=Thu, 19-Jan-2017 02:01:34
GMT;path=/

GET /theme/assets/img/content/~/media/400x265/students/23367_400x265.ashx HTTP/1.1
```

Change to Hex Dump Mode and view the payloads in raw Hex, as shown below.



```
00000220  34 38 34 37 35 39 38 32  38 3b 20 6e 61 70 69 65 48475982 8; napie
00000230  72 41 63 63 65 70 74 43  6f 6f 6b 69 65 73 3d 74 rAcceptC ookies=t
00000240  72 75 65 3b 20 5f 67 61  74 3d 31 3b 20 41 53 50 rue; _ga t=1; ASP
00000250  2e 4e 45 54 5f 53 65 73  73 69 6f 6e 49 64 3d 6c .NET_Ses sionId=l
00000260  77 31 31 6f 34 79 32 73  72 79 70 6f 64 31 72 71 w11o4y2s rypod1rq
00000270  62 34 34 79 68 7a 79 3b  20 73 63 5f 65 78 70 76 b44yhzy;  sc_expv
00000280  69 65 77 3d 30 0d 0a 43  6f 6e 6e 65 63 74 69 6f iew=0..C onnectio
00000290  6e 3a 20 6b 65 65 70 2d  61 6c 69 76 65 0d 0a 49 n: keep- alive..I
000002A0  66 2d 4d 6f 64 69 66 69  65 64 2d 53 69 6e 63 65 f-Modifi ed-Since
000002B0  3a 20 54 68 75 2c 20 30  34 20 46 65 62 20 32 30 : Thu, 0 4 Feb 20
000002C0  31 36 20 30 38 3a 30 38  3a 31 38 20 47 4d 54 0d 16 08:08 :18 GMT.
000002D0  0a 43 61 63 68 65 2d 43  6f 6e 74 72 6f 6c 3a 20 .Cache-C ontrol:
000002E0  6d 61 78 2d 61 67 65 3d  30 0d 0a 0d 0a          max-age= 0....
    00000000  48 54 54 50 2f 31 2e 31  20 33 30 34 20 4e 6f 74 HTTP/1.1  304 Not
    00000010  20 4d 6f 64 69 66 69 65  64 0d 0a 43 61 63 68 65  Modifie d..Cache
    00000020  2d 43 6f 6e 74 72 6f 6c  3a 20 70 72 69 76 61 74 -Control : privat
    00000030  65 2c 20 6d 61 78 2d 61  67 65 3d 36 30 34 38 30 e, max-a ge=60480
    00000040  30 0d 0a 45 78 70 69 72  65 73 3a 20 57 65 64 2c 0..Expir es: Wed,
    00000050  20 32 35 20 4a 61 6e 20  32 30 31 37 20 31 38 3a  25 Jan  2017 18:
    00000060  30 31 3a 33 34 20 47 4d  54 0d 0a 4c 61 73 74 2d 01:34 GM T..Last-
    00000070  4d 6f 64 69 66 69 65 64  3a 20 54 68 75 2c 20 30 Modified : Thu, 0
    00000080  34 20 46 65 62 20 32 30  31 36 20 30 38 3a 30 38 4 Feb 20 16 08:08
    00000090  3a 31 38 20 47 4d 54 0d  0a 53 65 72 76 65 72 3a :18 GMT. .Server:
    000000A0  20 4d 69 63 72 6f 73 6f  66 74 2d 49 49 53 2f 38  Microso ft-IIS/8
    000000B0  2e 35 0d 0a 53 65 74 2d  43 6f 6f 6b 69 65 3a 20 .5..Set- Cookie:
    000000C0  73 63 5f 65 78 70 76 69  65 77 3d 30 3b 20 70 61 sc_expvi ew=0; pa
```

Close the popup window. Wireshark now only shows the packets from the selected TCP Stream. You should be able to identify the 3-way handshake easily now.



Filter: tcp.stream eq 1

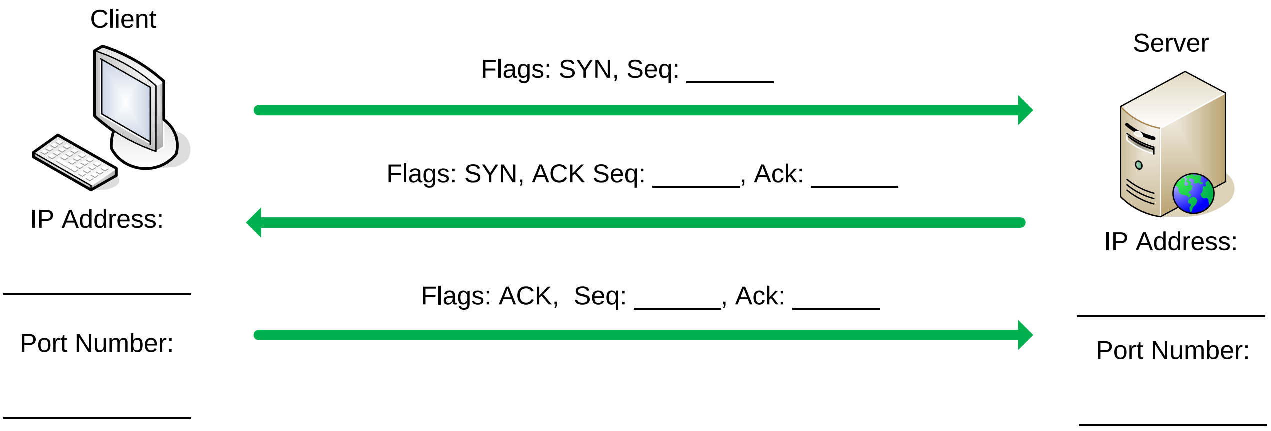
```
No.   Time          Source         Destination    Protocol  Length  Info
    3 0.270047000   10.1.99.7      146.176.5.23   TCP           62 gpfs > http [SYN] Seq=0 Wir
    5 0.271270000   146.176.5.23   10.1.99.7      TCP           60 http > gpfs [SYN, ACK] Seq=
    6 0.271300000   10.1.99.7      146.176.5.23   TCP           54 gpfs > http [ACK] Seq=1 Ack
   93 1.270010000   10.1.99.7      146.176.5.23   HTTP         803 GET /~/media/400x400/campus
```

Note: Wireshark has automatically created a **display filter** to filter out this TCP conversation.
In this case: **tcp/stream eq 1**
With is the equivalent of something similar to:

(ip.addr eq 10.10.x.7 and ip.addr eq 146.176.5.23) and (tcp.port eq 80 and tcp.port eq 1191)

Questions
Q: From your Wireshark Capture, fill in the diagram below with the IP Addresses and Port Numbers for the Client and the Server

Q: For each packet in the TCP 3-way handshake, fill in the Sequence and Acknowledgement numbers, on the diagram below.



Saving Packet Captures

Often captures need to be saved to disc, for later analysis using Wireshark or as input to other analysis tools.
Check how many packets are in your current capture. The details of total packets, packets displayed are shown in the status bar.

Questions
Q: How many total packets are in your capture?

To save the capture, select **File->Save As** and save the trace. By default this creates a Wireshark **pcapng** file, or if you select **pcap** it produces a file many tools can read and write.

For example a tcpdump output file is in this pcap format and can be read into Wireshark or Snort for analysis. This saves all the captured packets to the file.

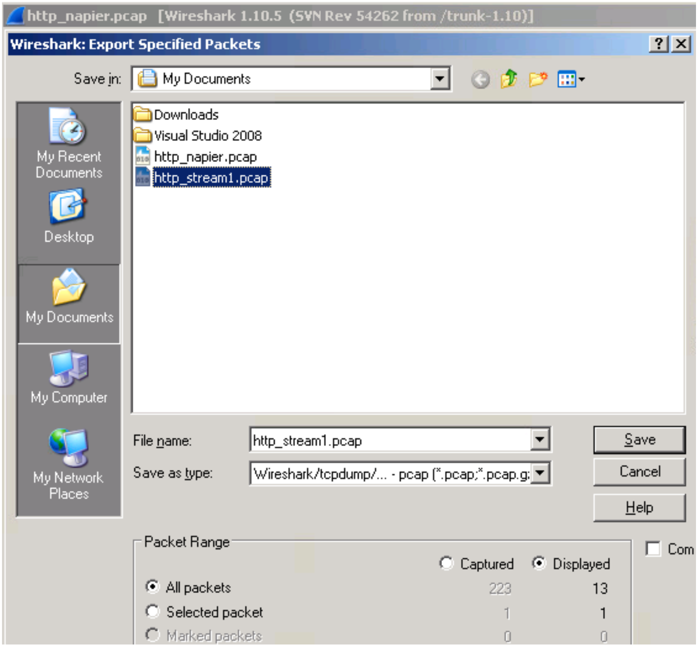
Questions

Q: Did you successfully save your capture to a .pcap file? (check folder) Yes/No

Q: Copy the **Display Filter** into the clipboard, and close and start Wireshark again, then reload the file. Was the whole capture saved or just the displayed packets?

Paste the display filter back into the Filter Bar, and Apply it.

To save *only the displayed packets*, select **File-> Export Specified Packets**, and make sure the **Displayed** radio button is selected rather than the **Captured** option. This creates a **pcap** file, with only the packets filtered by the current display filter.



http_napier.pcap [Wireshark 1.10.5 (SVN Rev 54262 from /trunk-1.10)]

Wireshark: Export Specified Packets

Save in: My Documents

Downloads
Visual Studio 2008
http_napier.pcap
http_stream1.pcap

File name: http_stream1.pcap

Save as type: Wireshark/tcpdump/... - pcap (*.pcap;*.pcap.g

Packet Range

	○ Captured	● Displayed
● All packets	223	13
○ Selected packet	1	1
○ Marked packets	0	0

This can be very useful if analysing particular sessions, protocols, or user related sets of traffic – e.g. an insider attack based on 1 user. Also is dealing with large pcap files filtering can take long times to process!

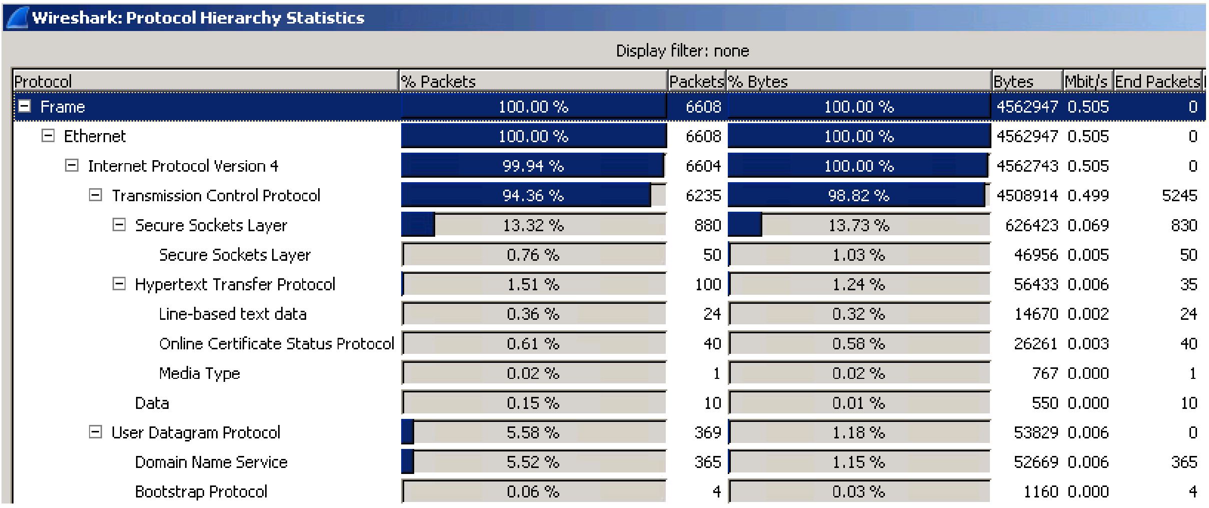
Questions

Q: Close and start Wireshark again, then reload the file. Was the whole capture saved or just the displayed packets?

Wireshark Statistics

Start a new capture, and generate some Web traffic by navigating to websites such as **www.schneier.com** and **www.napier.co.uk**, then stop the capture.

Select the **Statistics->Protocol Hierarchy** menu option. A window similar to the below should now display statistics about the **pcap**. Note that all the packets are Ethernet (Local Area Network) packets, but at the network layer most of the packets are TCP, but some are UDP.



Wireshark: Protocol Hierarchy Statistics

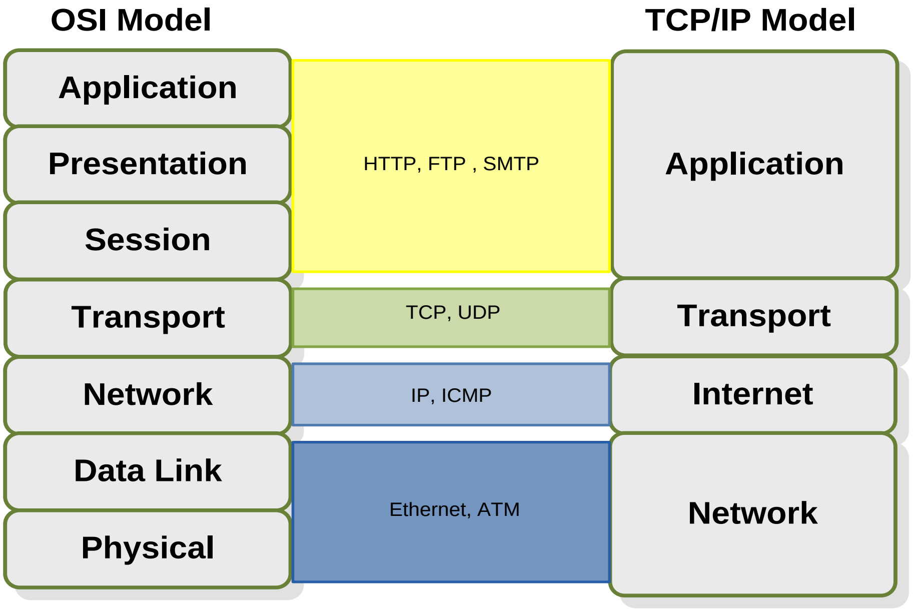
Display filter: none

Protocol	% Packets	Packets	% Bytes	Bytes	Mbit/s	End Packets
⊟ Frame	100.00 %	6608	100.00 %	4562947	0.505	0
⊟ Ethernet	100.00 %	6608	100.00 %	4562947	0.505	0
⊟ Internet Protocol Version 4	99.94 %	6604	100.00 %	4562743	0.505	0
⊟ Transmission Control Protocol	94.36 %	6235	98.82 %	4508914	0.499	5245
⊟ Secure Sockets Layer	13.32 %	880	13.73 %	626423	0.069	830
Secure Sockets Layer	0.76 %	50	1.03 %	46956	0.005	50
⊟ Hypertext Transfer Protocol	1.51 %	100	1.24 %	56433	0.006	35
Line-based text data	0.36 %	24	0.32 %	14670	0.002	24
Online Certificate Status Protocol	0.61 %	40	0.58 %	26261	0.003	40
Media Type	0.02 %	1	0.02 %	767	0.000	1
Data	0.15 %	10	0.01 %	550	0.000	10
⊟ User Datagram Protocol	5.58 %	369	1.18 %	53829	0.006	0
Domain Name Service	5.52 %	365	1.15 %	52669	0.006	365
Bootstrap Protocol	0.06 %	4	0.03 %	1160	0.000	4

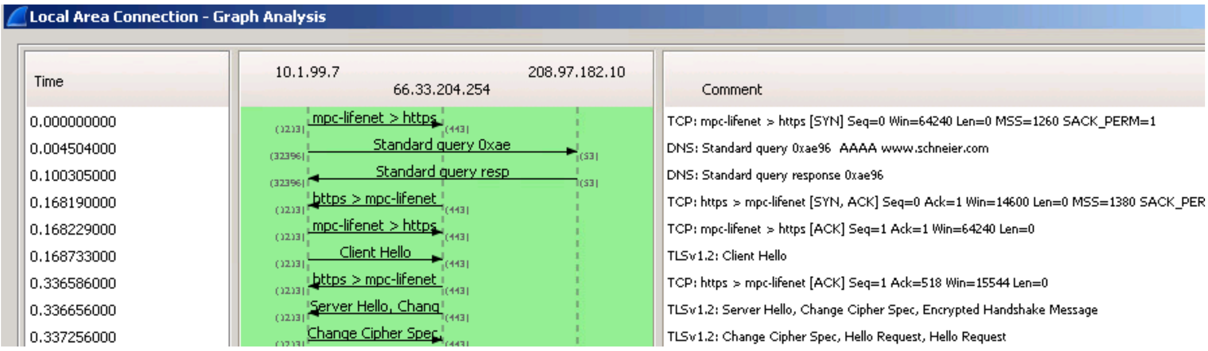
Questions

Q. What percentage of packets in your capture are TCP, and give an example of the higher level protocol which uses TCP?

Q. What percentage of packets in your capture are UDP, and give an example of the higher level protocol which uses UDP? (use the figure below)



Select the **Statistics->Flow Graph** menu option. Choose **General Flow** and **Network Source** options, and click the **OK** button. A window similar to the below should be displayed, showing a visual representation of the flow of traffic, including the timeline in the first column and a description of each packet in the last.


Local Area Connection - Graph Analysis

Time	10.1.99.7 66.33.204.254 208.97.182.10	Comment
0.000000000	(1213) mpc-lifenet > https (443)	TCP: mpc-lifenet > https [SYN] Seq=0 Win=64240 Len=0 MSS=1260 SACK_PERM=1
0.004504000	(32396) Standard query 0xae (53)	DNS: Standard query 0xae96 AAAA www.schneier.com
0.100305000	(32396) Standard query resp (53)	DNS: Standard query response 0xae96
0.168190000	(1213) https > mpc-lifenet (443)	TCP: https > mpc-lifenet [SYN, ACK] Seq=0 Ack=1 Win=14600 Len=0 MSS=1380 SACK_PER
0.168229000	(1213) mpc-lifenet > https (443)	TCP: mpc-lifenet > https [ACK] Seq=1 Ack=1 Win=64240 Len=0
0.168733000	(1213) Client Hello (443)	TLSv1.2: Client Hello
0.336586000	(1213) https > mpc-lifenet (443)	TCP: https > mpc-lifenet [ACK] Seq=1 Ack=518 Win=15544 Len=0
0.336656000	(1213) Server Hello, Chang (443)	TLSv1.2: Server Hello, Change Cipher Spec, Encrypted Handshake Message
0.337256000	(1213) Change Cipher Spec (443)	TLSv1.2: Change Cipher Spec, Hello Request, Hello Request

Filtering out Interesting Traffic using Wireshark.

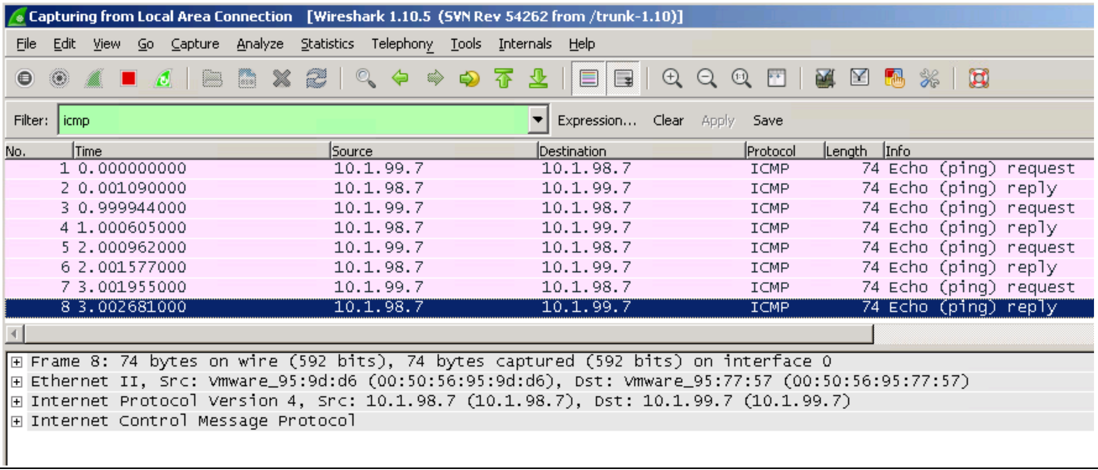
Start a new Wireshark capture. Generate some web traffic using your browser again. Open a Windows console window, and generate some ICMP traffic by using the **ping** command line tool to check the connectivity of the current network's default gateway. Generate some more web traffic.

Stop the capture and Wireshark.
The Address Resoloution Protocol (ARP) and ICMP packets generated by the ping command can be difficult to pick out manually especially on a busy network
We can create a **display filter** to only show ARP and ICMP packets and analyse those separately.

Some useful Wireshark **display filters** can be found at:
http://wiki.wireshark.org/DisplayFilters
https://www.wireshark.org/docs/dfref/

To filter by protocol we can simply enter the protocol into the **display Filter** box as we did with the IP Address earlier (or add to the IP Address filter with an **and** if you are on a shared/busy network).

Try entering ICMP into the **Filter** box, and press RETURN.
Wireshark should filter and display only the ICMP protocol packets, such as shown below:


Capturing from Local Area Connection [Wireshark 1.10.5 (SVN Rev 54262 from /trunk-1.10)]

Filter: icmp

```
No.  Time          Source       Destination  Protocol Length Info
   1 0.000000000   10.1.99.7    10.1.98.7    ICMP       74 Echo (ping) request
   2 0.001090000   10.1.98.7    10.1.99.7    ICMP       74 Echo (ping) reply
   3 0.999944000   10.1.99.7    10.1.98.7    ICMP       74 Echo (ping) request
   4 1.000605000   10.1.98.7    10.1.99.7    ICMP       74 Echo (ping) reply
   5 2.000962000   10.1.99.7    10.1.98.7    ICMP       74 Echo (ping) request
   6 2.001577000   10.1.98.7    10.1.99.7    ICMP       74 Echo (ping) reply
   7 3.001955000   10.1.99.7    10.1.98.7    ICMP       74 Echo (ping) request
   8 3.002681000   10.1.98.7    10.1.99.7    ICMP       74 Echo (ping) reply
```

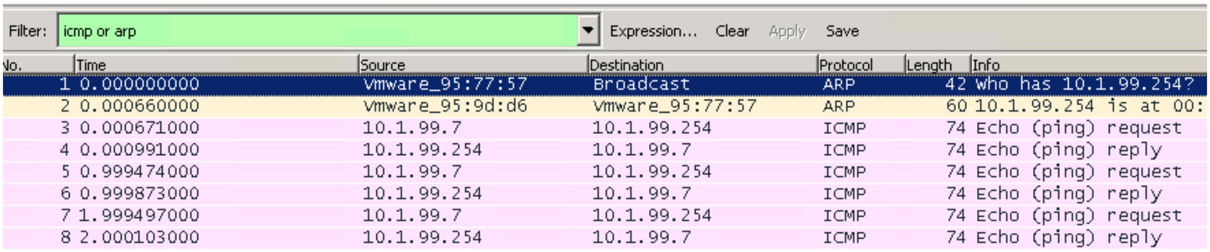
```
⊞ Frame 8: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0
⊞ Ethernet II, Src: Vmware_95:9d:d6 (00:50:56:95:9d:d6), Dst: Vmware_95:77:57 (00:50:56:95:77:57)
⊞ Internet Protocol Version 4, Src: 10.1.98.7 (10.1.98.7), Dst: 10.1.99.7 (10.1.99.7)
⊞ Internet Control Message Protocol
```

Questions

Q: How many ICMP packets are being displayed?

Try filtering out TCP, then UDP, then HTTP.

We can also build more complex filters which combine simpler filters. Try **ICMP or ARP** to shown packets of either protocol. i.e. all the packets generated by out ping tool.



Filter: icmp or arp

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	Vmware_95:77:57	Broadcast	ARP	42	Who has 10.1.99.254?
2	0.000660000	Vmware_95:9d:d6	Vmware_95:77:57	ARP	60	10.1.99.254 is at 00:
3	0.000671000	10.1.99.7	10.1.99.254	ICMP	74	Echo (ping) request
4	0.000991000	10.1.99.254	10.1.99.7	ICMP	74	Echo (ping) reply
5	0.999474000	10.1.99.7	10.1.99.254	ICMP	74	Echo (ping) request
6	0.999873000	10.1.99.254	10.1.99.7	ICMP	74	Echo (ping) reply
7	1.999497000	10.1.99.7	10.1.99.254	ICMP	74	Echo (ping) request
8	2.000103000	10.1.99.254	10.1.99.7	ICMP	74	Echo (ping) reply

Note the results in Wireshark. The initial ARP request broadcast from your system determining the physical MAC address of the network IP Address of the gateway, via the ARP reply from the system. After the ARP request, the pings (ICMP echo request and replies) can be seen.

Questions

Q: After the first ping, are the ARP and ICMP packets captured by Wireshark?

Yes/No

Q: Try more pings while viewing the filtered traffic in Wireshark. After a second or third ping, is the ARP and ICMP packets captured by Wireshark?

ARP: Yes/No
ICMP: Yes/No

Q: Why is this?

If pinging the same system more than once, to force ARP apckets to be generated, delete the ARP cache on your system, using the Windows `arp` command, as shown below, so an new ARP request will be generated.

```
C:\> arp -d *
```

Then check with:

```
C:\> arp -a
```

Q: What are the MAC addresses associated with the firewall?

Check on the firewall via the web app.

Now delete the IP address on the DMZ gateway interface on the firewall, and reassess:

Q: Can you ping the 10.10.x.7 port from the host on the 10.10.y.7? Yes/No

Q: Can you ping the 10.10.*x*.7 port from the host on the 10.10.y.7? Yes/No

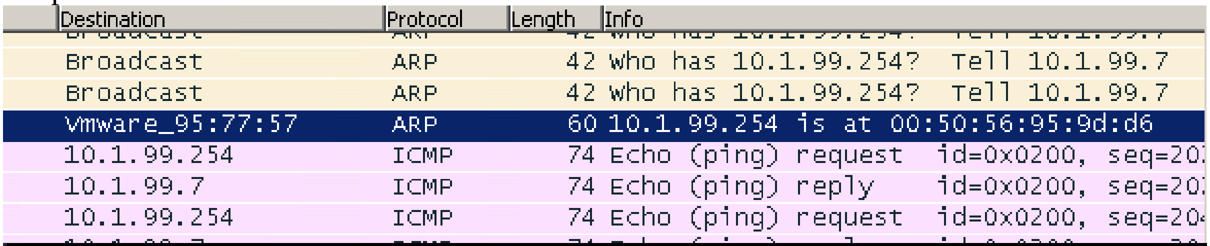
Now run Wireshark on both your hosts, and repeat. Examine you network trace, and determine the unsuccessful ping request, and ping reply.

Q: Which ICMP type codes are used for the request and the unsuccessful reply?

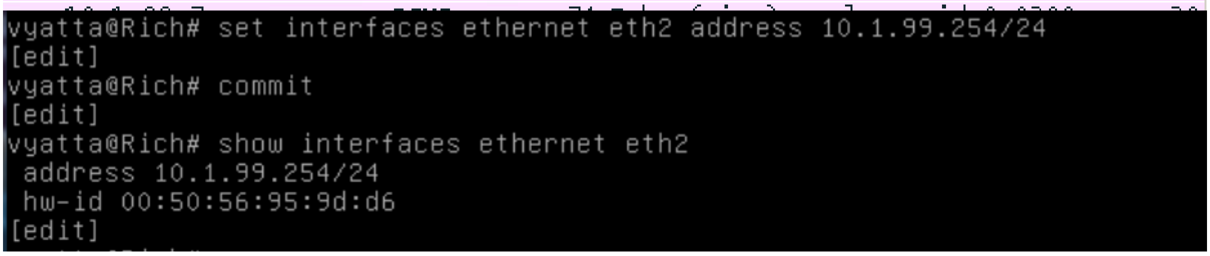
Now, reapply the IP address, and determine the MAC addresses of the gateway adapter from Wireshark, and check this against the configuration of the firewall.

Q: What are the MAC addresses associated with the firewall?

Outputs similar to:



```
    Destination              Protocol   Length  Info
    Broadcast                ARP          42 Who has 10.1.99.254?  Tell 10.1.99.7
    Broadcast                ARP          42 Who has 10.1.99.254?  Tell 10.1.99.7
    Broadcast                ARP          42 Who has 10.1.99.254?  Tell 10.1.99.7
    Vmware_95:77:57          ARP          60 10.1.99.254 is at 00:50:56:95:9d:d6
    10.1.99.254              ICMP         74 Echo (ping) request  id=0x0200, seq=20
    10.1.99.7                ICMP         74 Echo (ping) reply    id=0x0200, seq=20
    10.1.99.254              ICMP         74 Echo (ping) request  id=0x0200, seq=20
```



```
vyatta@Rich# set interfaces ethernet eth2 address 10.1.99.254/24
[edit]
vyatta@Rich# commit
[edit]
vyatta@Rich# show interfaces ethernet eth2
 address 10.1.99.254/24
 hw-id 00:50:56:95:9d:d6
[edit]
```

Now with a browser on each host, access the Web server on the other network.

Q: Can you access the Web server on the 10.10.x.7 from 10.10.*x*.7? Yes/No

Q: Can you access the Web server on the 10.10.*x*.7 from 10.10.x.7? Yes/No

As before, disable the IP address on the eth1 port, and reapply (make sure you refresh the cache on the browser):

Q: Can you access the Web server on the 10.10.x.7 from 10.10.*x*.7? Yes/No

Q: Can you access the Web server on the 10.10.*x*.7 from 10.10.x.7? Yes/No

Reapply IPs as before, and test that all systems have connectivity.

Network Scanning for Hosts & Services

Check the help for the nmap network scanning tool again – use **`man nmap`** and **`nmap -h | more`**

Q: Which flag specifies a host discovery scan only?

Q: What does the –n flag `do?`

Startup Wireshark on each of your hosts and capture traffic. From Linux you can use:

```
sudo wireshark&
```

From the Windows host, in a command window, run an **nmap host discovery scan** on the entire subnet the Linux box is on:

```
nmap -sP -n 10.10.x.0/24
```

Q: How many hosts did nmap scan for? How many did it report finding?

Q: What type of packets were sent to discover hosts?
(run for your single host if confusing)

From the Windows host , run an **nmap port scan** on the subnet the Linux box is on:

```
nmap -sS -n 10.10.x.0/24
```

Q: List some well known ports are open/services are running on the Linux host?
Ports: Protocol: Services:

Run a similar nmap **host discovery scan** and **port scan** from the Linux host to the Windows network/ systems found.
]
Q: List some of the common ports are open/ services are running on the Windows host?
Ports: Protocol: Services:

Q: Review the scan packets with Wireshark. What type of packets were sent to individual ports by nmap during the scans?

Check the results with the Ubuntu/Windows commands which list a hosts running network services. `ss -l` and `netstat -ap TCP` respectively.