

## Lab 1.5

# Network traffic analysis and wireless network security

### Objectives

After completing this lab, you will be able to:

- Perform network analysis using a packet sniffer.
- Realize the importance of network security, especially in a wireless network environment.

### Synopsis

*Network analysis* (also known as *traffic analysis*, *protocol analysis*, *packet analysis*, *packet sniffing* and so on) is the process of capturing network traffic and examining it closely in order to deduce information from it. A *packet sniffer* (also known as a *network analyser*, *protocol analyser* or *network protocol analyser*) is a hardware device or software that captures, logs and analyses network traffic. Network analysis can be used for both good and evil. The network administrator performs network analysis to monitor network usage, analyse network problems and debug client-server communications. The network security practitioner performs it to detect network intrusion attempts and violations against network usage policies. Hackers perform it to gain information for effecting network intrusion and other unauthorized (and often illegal) activities. An engineer (or a student like you) can also use network analysis to investigate, study and reverse engineer protocols used over the network. In this lab, you will use a powerful and free packet sniffer, *Wireshark*, to capture and analyse the traffic between your PC and selected remote hosts on the Internet. In addition, you will use a wireless networking tool, *NetStumbler*, to detect the wireless LANs in your neighbourhood. You will be surprised that many people are not running their wireless LANs in secure mode and hence are vulnerable to wireless sniffing.

## Warning

Before you rush to acquire and install a packet sniffer, you are warned that unauthorized sniffing can lead to serious consequences. Do not just install and use a packet sniffer on your company or school network, even though you think you are just 'experimenting'! While it is fine to sniff the network traffic originating from your own PC at home, it may not be so elsewhere. If you are not performing this lab at home, be sure to ask for explicit authorization from the network administrator or your instructor before proceeding with this lab.

## Prerequisites

- Your PC is connected to the Internet.
- Your PC is running Microsoft Windows XP or later.
- You know which network interface card on your PC is used to access the Internet.
- To perform Step 5, you need to have a wireless network card installed and configured properly on your PC such that it can access the wireless network in your environment.

## Expected duration

Approximately 60–90 minutes

## Procedure

### Step 1 Downloading and installing Wireshark

Open the Web browser on your PC and visit the Wireshark website:

<http://www.wireshark.org/>

Download and install the latest version of Wireshark. As of March 2009, the latest version is 1.06. The file size of the installer is about 21 MB. With broadband Internet access, you should be able to download it in less than five minutes under normal traffic conditions.



**Figure 1.5.1**

The installation of Wireshark is straightforward. Simply accept the default options and it should install without any problem. During the installation process, ensure that *WinPcap*,<sup>1</sup> the Windows Packet Capture Library, is selected to be installed, although it should have been selected by default. In fact, it is WinPcap that performs the actual packet capturing and filtering, and Wireshark is just the GUI front-end utilizing WinPcap. Without WinPcap, Wireshark will not be able to capture network traffic at all. The whole installation process should take around one minute.

<sup>1</sup> For more information on WinPcap, visit the website <http://www.winpcap.org>.

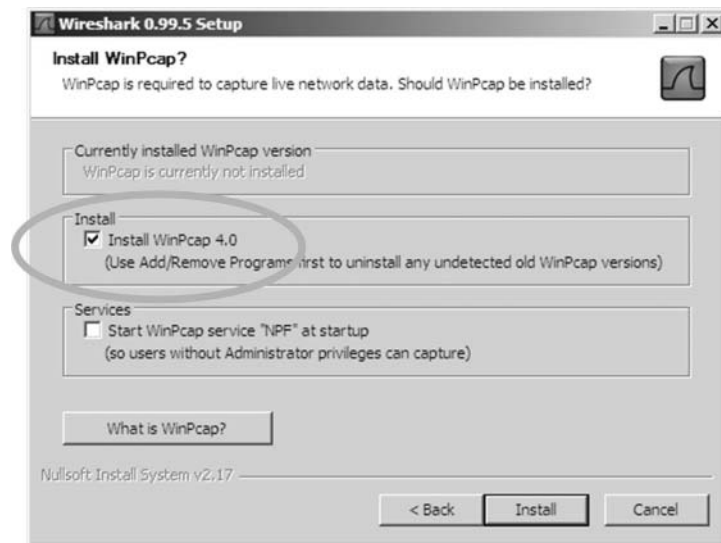


Figure 1.5.2

It is also recommended that you download the *Wireshark User's Guide* for your reference. It can be found here:

<http://www.wireshark.org/docs>

## Step 2 Getting started with Wireshark

Click **Start** → **Programs** → **Wireshark** → **Wireshark** to launch the Wireshark program. Wireshark opens with an empty main window as below. Now let's start a simple capture. Click **Interfaces** on the **Capture** menu.

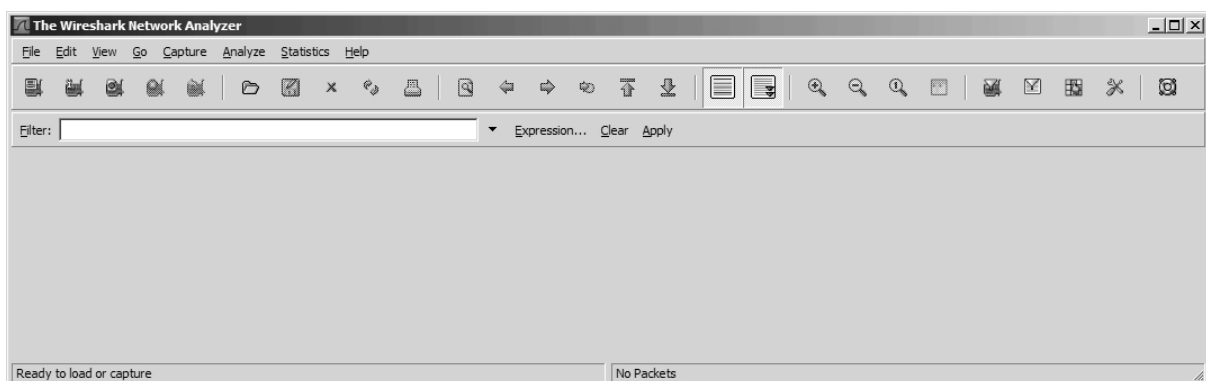
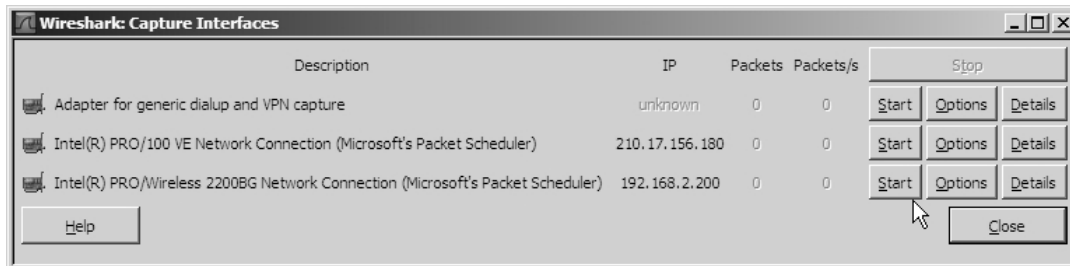


Figure 1.5.3

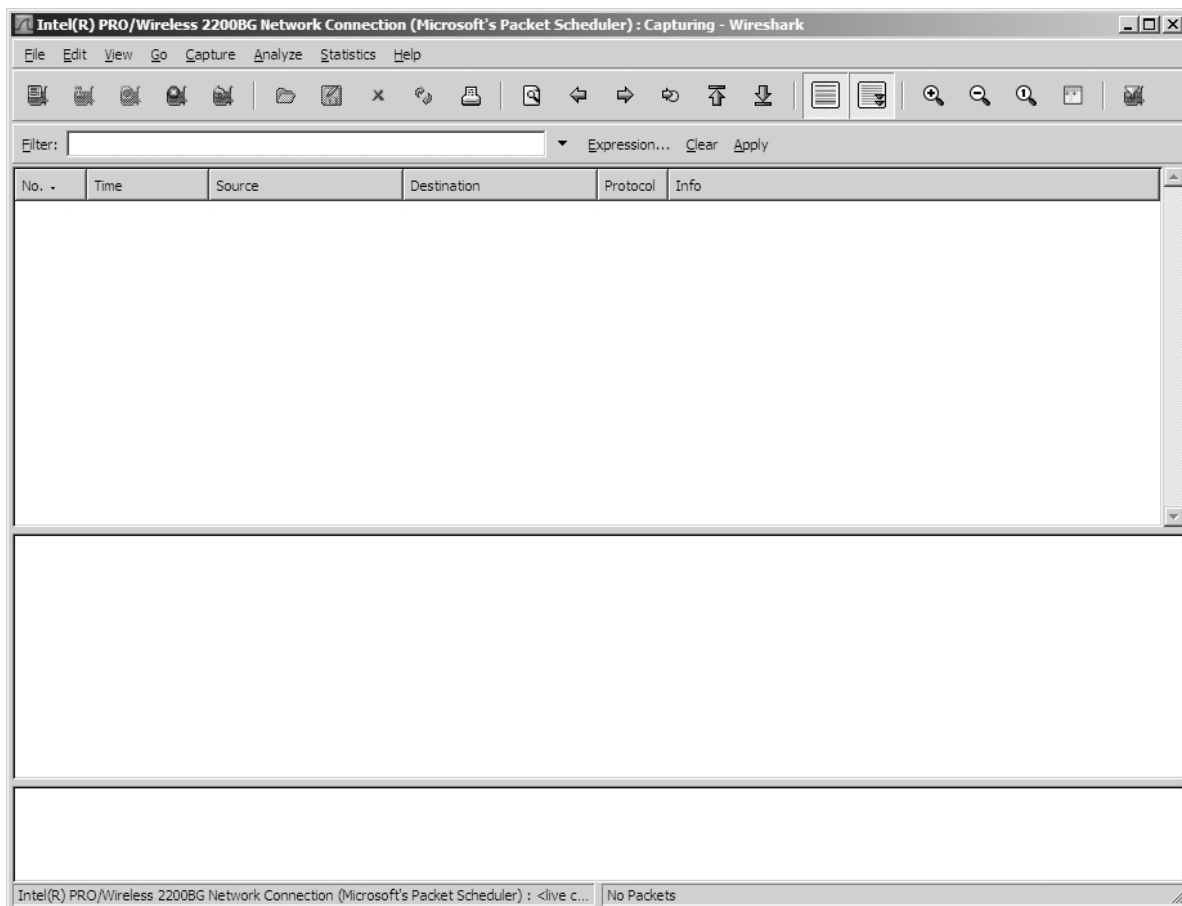
The **Wireshark: Capture Interfaces** window appears, showing the network interface cards available on your PC. The figure below shows the network interface cards available on our PC. Yours will be different from ours.

Our PC uses the Intel PRO/Wireless 2200BG network interface to access the Internet. Identify the network interface card on your PC that is used to access the Internet. Click the corresponding **Start** button on the right.



**Figure 1.5.4**

Wireshark begins capturing on the selected network interface card and the main window becomes as shown in the following figure.



**Figure 1.5.5**

Next, perform the following tasks that will access the Internet:

- Open the Web browser and visit google.com. Search for *Wireshark*, *Ethereal* and *packet sniffing* one by one. Visit some of the websites in the search results.
- Check your email using a POP3 client such as Outlook Express or Mozilla Thunderbird.
- Perform some DNS queries using *nslookup*.

As you perform these tasks, packets are generated and received by your network interface card and these packets are captured by Wireshark. Notice that the main window is updated dynamically as capturing is in process. In the figure below, the various main window components are labelled on the left.

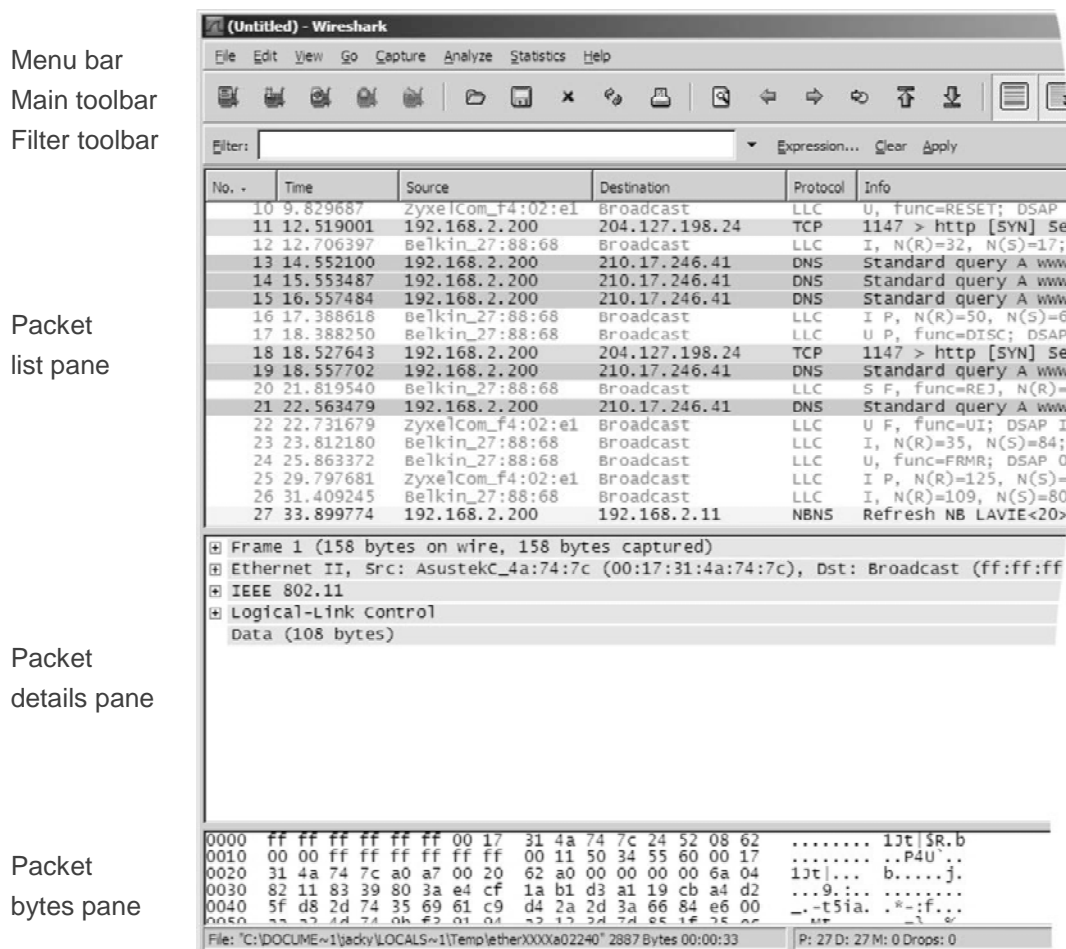


Figure 1.5.6

The following list, adapted from the *Wireshark User's Guide*, briefly describes the main window components (you may want to refer to the guide for more details):

- The *menu bar*, which is common in GUI programs, contains menu items that are used to start actions.

- The *main toolbar* provides quick access to frequently used items from the menu.
- The *filter toolbar* provides a way to directly manipulate the currently used display filter. (Display filters will be described in the next step.)
- The *packet list pane* displays a summary of each packet captured. By clicking on packets in this pane, you control what is displayed in the packet details pane and the packet bytes pane.
- The *packet details pane* displays the packet selected in the packet list pane in more detail.
- The *packet bytes pane* displays the data from the packet selected in the packet list pane, with the field selected in the packet details pane highlighted.

When you have finished the above tasks, click **Stop** on the **Capture** menu or simply press **Ctrl+E** to stop capturing. Scroll through the packet list in the packet list pane to find the packets corresponding to your Google searches. Click on some of the packets in the packet list pane and examine the details shown in the packet details pane. Select and expand the displayed protocols and fields in the packet details pane one by one. Examine the information shown in the packet details pane and packet bytes pane. Do you understand the information disclosed? Can you relate the information to your knowledge of the TCP/IP protocol suite?

Scroll through the packet list again to identify the packets corresponding to your POP3 session and DNS queries. Examine the information shown in the packet details pane for each of the corresponding packets. Again, try to relate the information to your knowledge of the TCP/IP protocol suite.

After finishing the investigation, save the capture. Click **Save As** on the **File** menu to open the **Save file as** dialog box. Supply a file name and, optionally, choose an appropriate folder to save your capture.

### Step 3 Using display filters in Wireshark

You may have noticed in the previous step that a Wireshark capture can easily produce a lot of information extraneous to what we are interested in. In order to make it easier to find the packets that we are interested in, we can set up a display filter to specify which packets are shown in the packet list pane.<sup>2</sup> The following are a few examples of useful display filters for you to try out, to display only:

- HTTP traffic: `http`
- ICMP traffic: `icmp`
- HTTP / DNS traffic: `http / dns`
- traffic on tcp port 12345: `tcp.port == 12345`
- those packets to or from the IP address 192.168.2.10: `ip.addr == 192.168.2.10`

<sup>2</sup> Alternatively, we may set up a capture filter to restrict Wireshark to capture only the packets that we are interested in. Unfortunately, the syntaxes of capture filters and display filters are slightly different. We will only discuss display filters in this lab. Refer to the *Wireshark User's Guide* for details on capture filters.

- those packets to or from the host `www.gmail.com`: `ip.addr == www.gmail.com`
- HTTP packets to or from the host `www.gmail.com`: `http.host == www.gmail.com`

These simple filters are enough to get you started. To learn more about Wireshark display filters, refer to the *Wireshark User's Guide* and the following online manual page:

<http://www.wireshark.org/docs/man-pages/wireshark-filter.html>

If you closed Wireshark after finishing Step 1, launch Wireshark again and open the capture file that you saved. Otherwise, the capture should still be displayed in the main window. Locate the filter toolbar and the filter text box. Type `http` in the text box. Press the **Enter** key or click **Apply** on the far right of the filter toolbar.

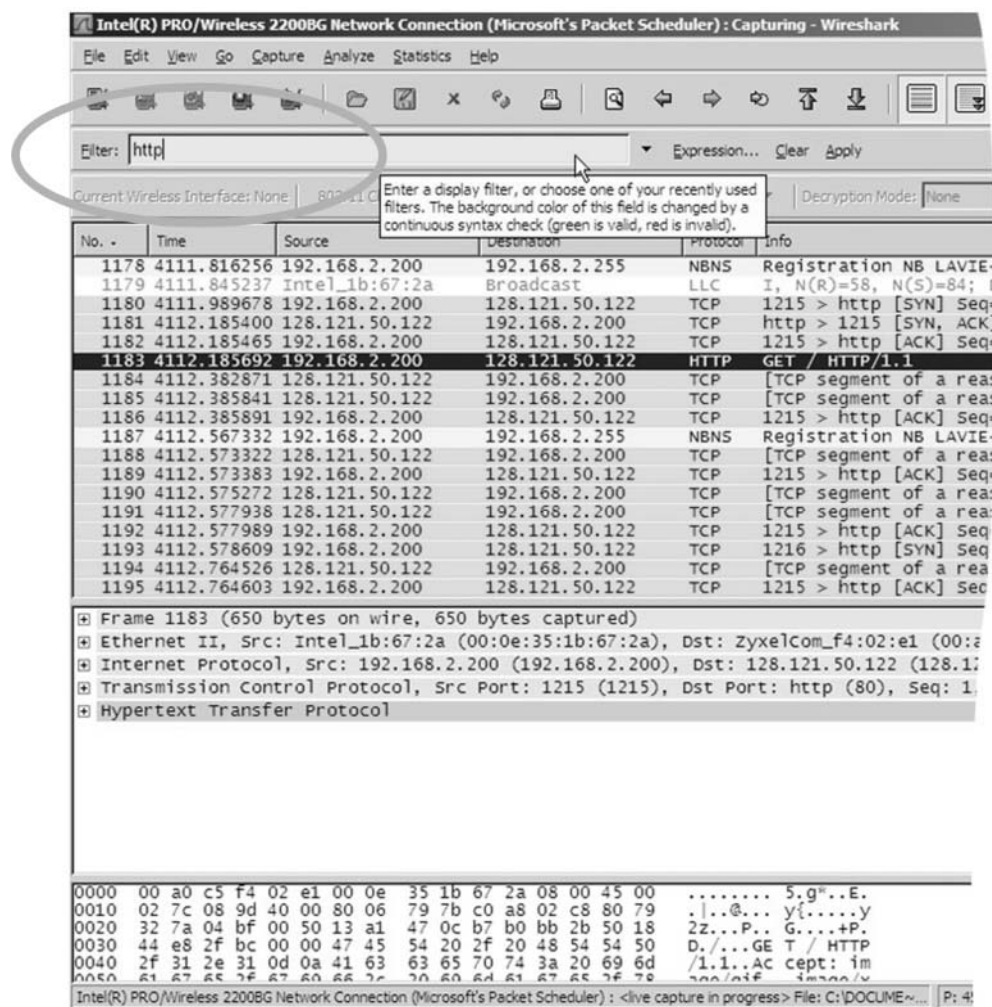
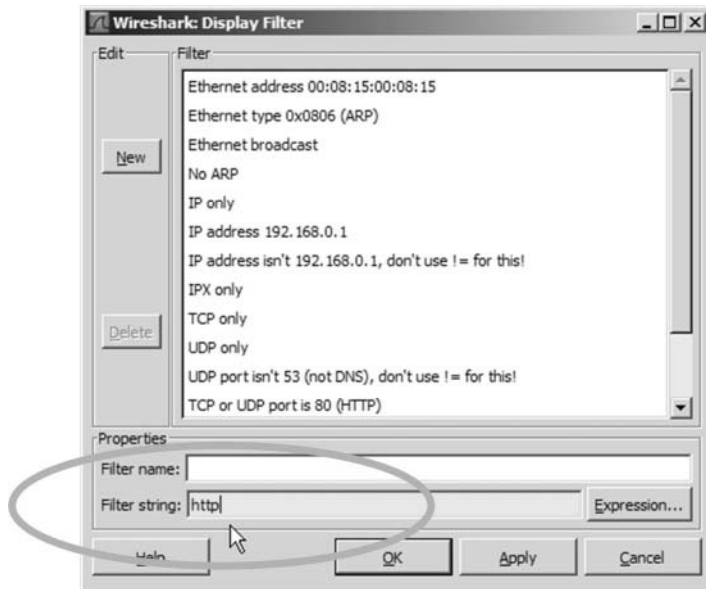


Figure 1.5.7



Alternatively, click **Display Filters** on the **Analyze** menu. The **Display Filter** dialog box appears, in which you can define a new display filter or edit existing ones. Type *http* in the **Filter string** text box and press the **OK** button to apply the filter.



**Figure 1.5.8**

After the display filter has been applied, the main window will show only HTTP packets.

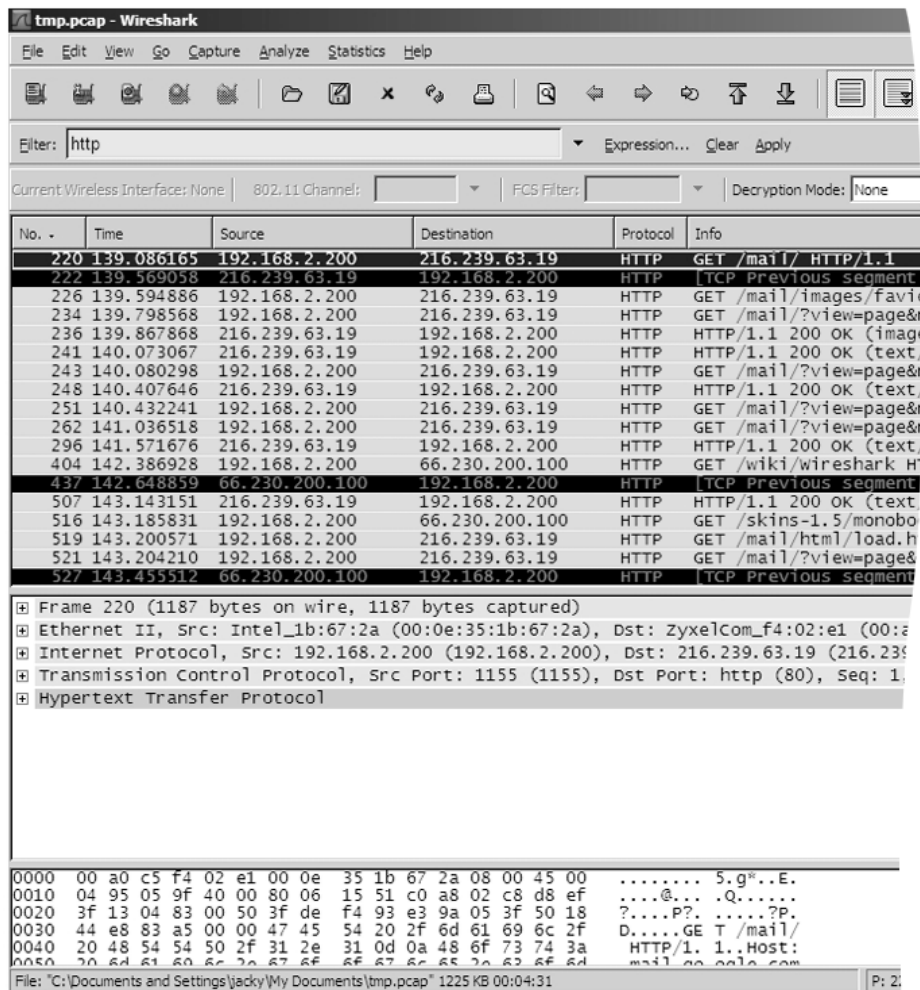


Figure 1.5.9

Change the filter to `http.host == www.google.com` to further restrict the displayed packets. Now it should be very easy to locate the Google searches that you have performed in the previous step.

Experiment with different filters to see the effect on the list of displayed packets.

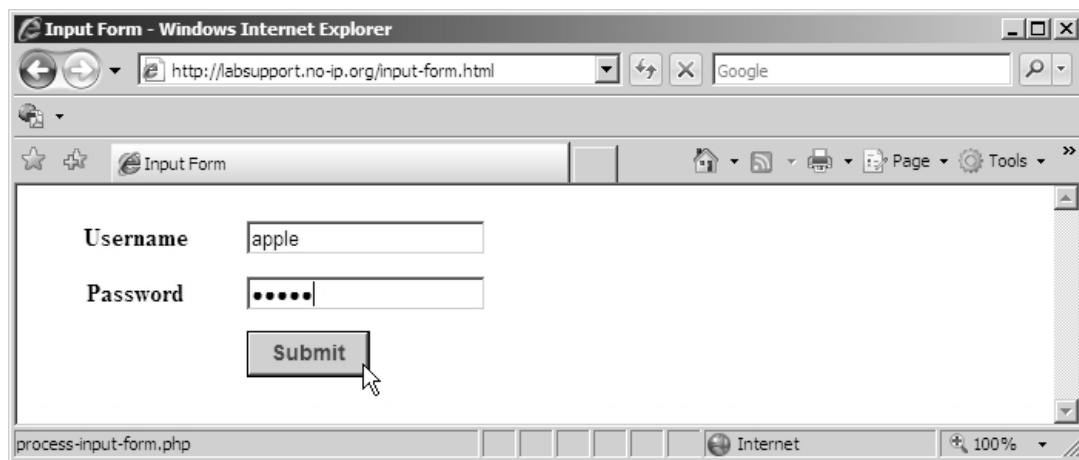
#### Step 4 Sniffing the information submitted via a webpage

In this step, you will use Wireshark to sniff the HTTP traffic between your PC and the website at `labupport.no-ip.org`. You will find out that information submitted via an unencrypted webpage can be easily revealed by Wireshark or a similar packet sniffer. As such, you should never submit any sensitive information via an unencrypted webpage.

If your previous capture is still open, close it now by clicking **Close** on the **File** menu or simply press **Ctrl+W**. Start a new capture as described in Step 2. Launch your browser and visit the following URL:

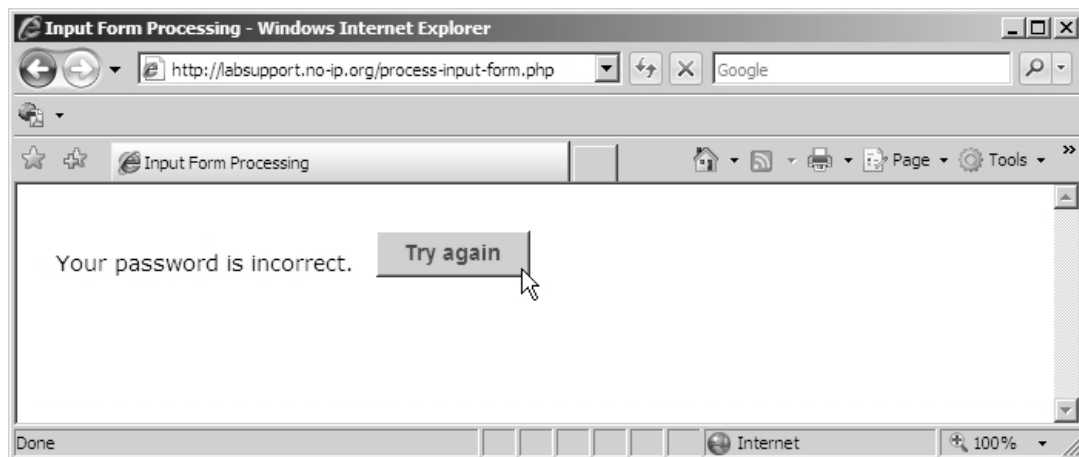
`http://labsupport.no-ip.org/input-form.html`

On the input form, fill in any name in the **Username** field and **123** in the **Password** field. Then click the **Submit** button.



**Figure 1.5.10**

Since the password is incorrect, the following output will be displayed.



**Figure 1.5.11**

Click the **Try again** button to navigate back to the original input form. Now change the password to **12345** and submit it again. The following greeting will be displayed.

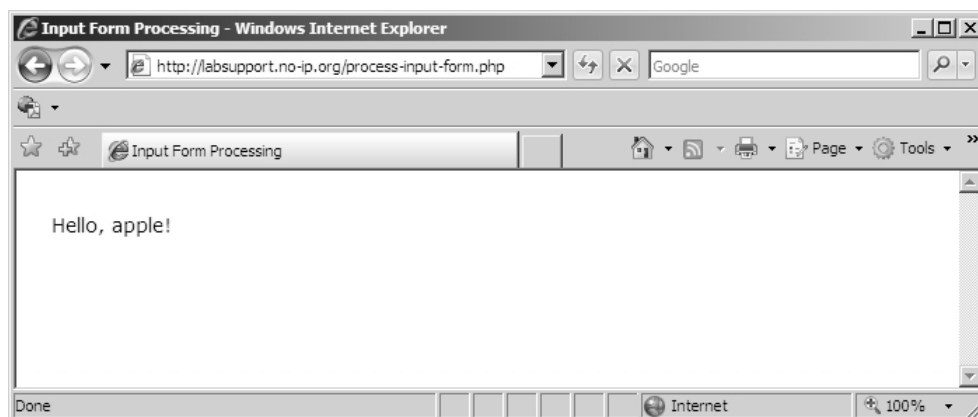


Figure 1.5.12

Press **Ctrl+E** to stop the Wireshark capture. Apply the display filter `http.host == labsupport.no-ip.org`. The main window should now show captured HTTP packets transmitted to and from `labsupport.no-ip.org`.

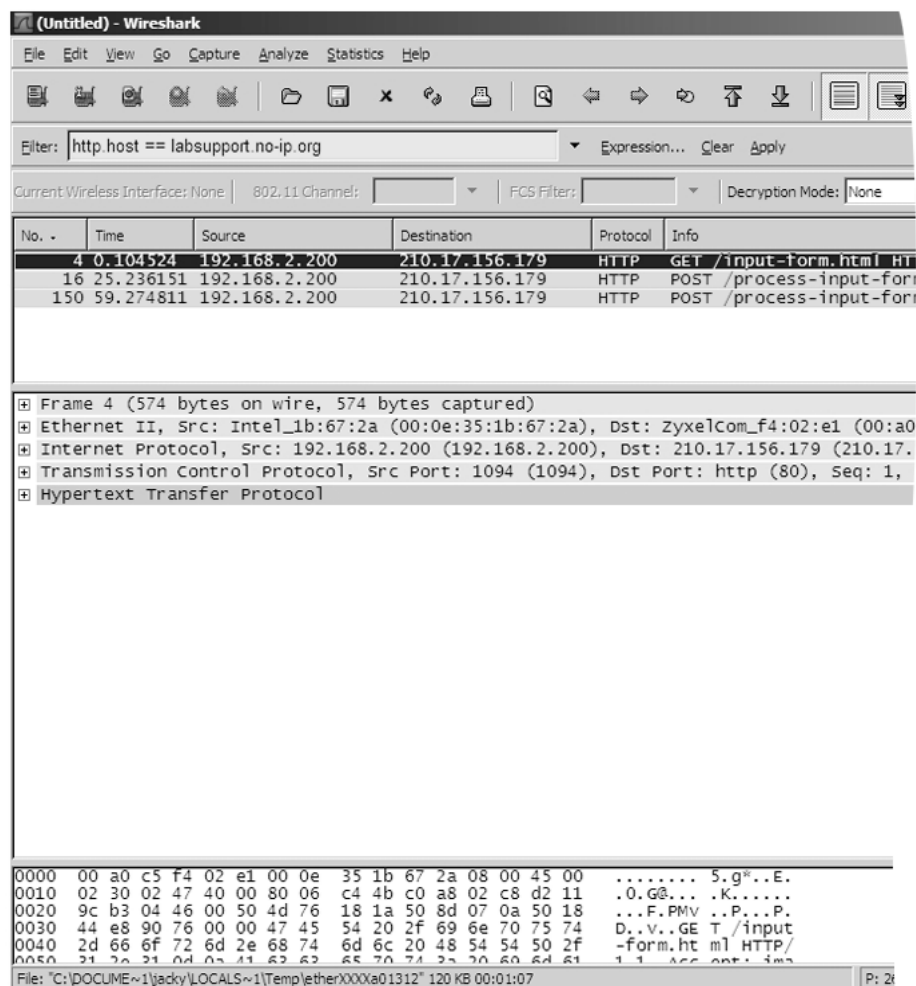


Figure 1.5.13

Select the last HTTP packet in the packet list pane. In the packet details pane, select **Line-based text data** and expand it. Can you identify the username and password that you submitted via the Web input form? Are they comprehensible to you?

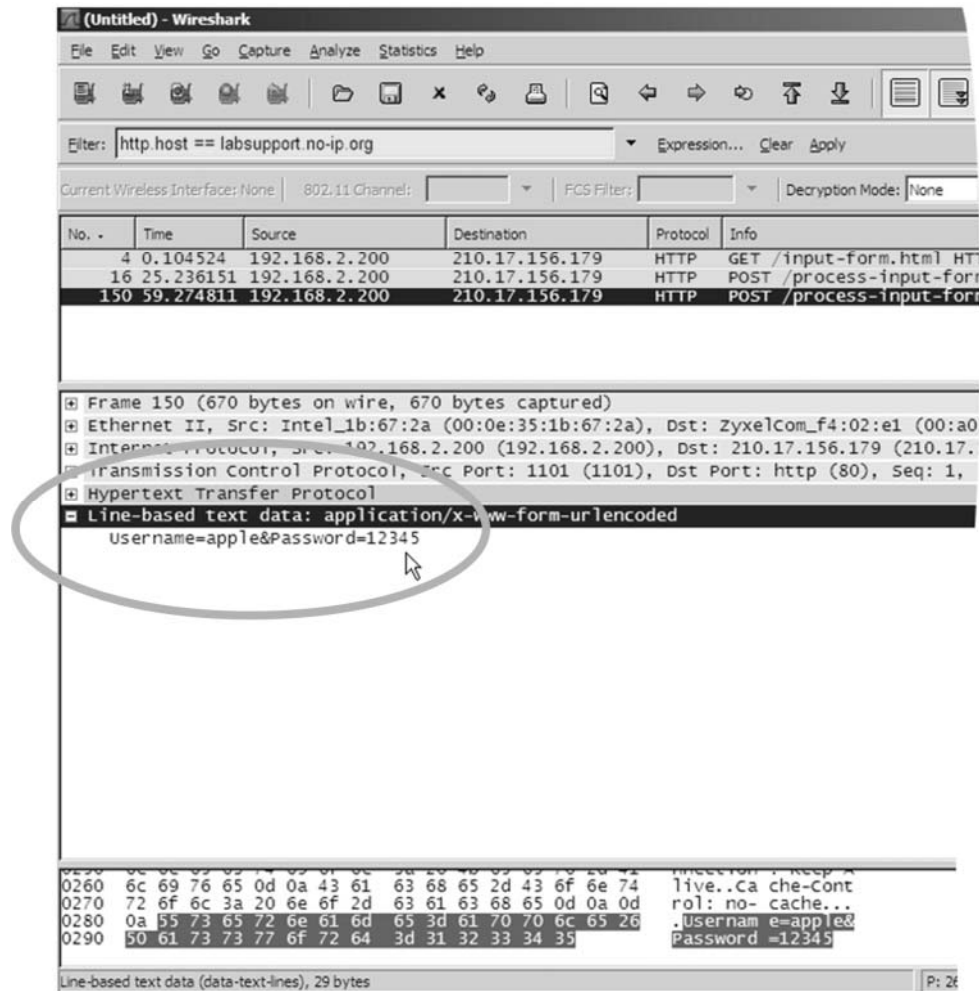


Figure 1.5.14

You can see that sensitive information transmitted via an unencrypted webpage can be easily identified. Start another capture. Access an SSL-encrypted input form on some other website, such as your e-banking login page. Log in as usual and then stop the capture. Repeat the above investigation. Can you identify the username and password this time? Are they comprehensible at all?

## Step 5 Wireless sniffing (optional)

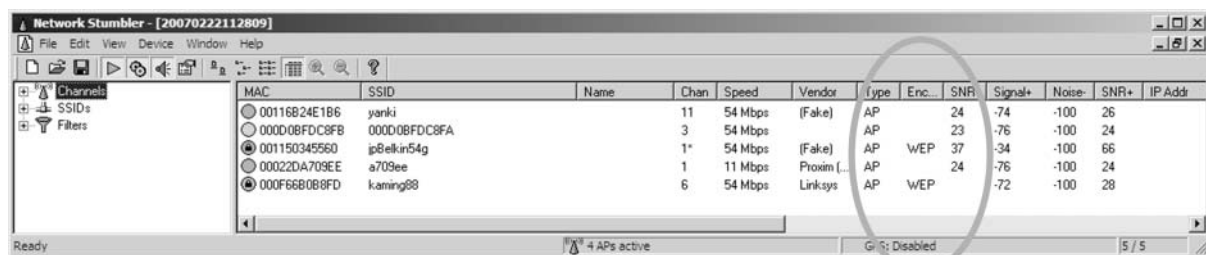
You can only perform this step if there is a wireless network interface card installed on your PC. It is assumed that you have already configured the wireless network interface card properly so that it can access the wireless network in your environment.

In the last step, you saw that sending sensitive information over an unencrypted communication channel is very dangerous. When the unencrypted communication channel is a conventional wired LAN, the intruder has to physically attach the sniffing device to your LAN, which is not always possible. However, in a wireless LAN environment, there is no such need. In the following, you will install and run *NetStumbler*, a wireless networking tool that allows you to detect wireless LANs in your neighbourhood. Using NetStumbler, you can easily find out which wireless LANs are running in insecure mode and hence are vulnerable to packet sniffing.

Open the Web browser on your PC and visit the NetStumbler website:

<http://www.netstumbler.com/>

Download and install the latest version of NetStumbler. As of February 2009, the latest version is 0.4. The file size of the installer is only 1.3 MB and it should be downloaded in a few seconds. The installation is straightforward and also takes only a few seconds. When the installation is finished, launch NetStumbler. Let it run for a few minutes and you will see an output similar to the one below.



MAC	SSID	Name	Chan	Speed	Vendor	Type	Enc.	SNR	Signal+	Noise-	SNR+	IP Addr
00116B24E186	yanki		11	54 Mbps	(Fake)	AP		24	-74	-100	26	
000D08FDC8F8	000D08FDC8FA		3	54 Mbps		AP		23	-76	-100	24	
001150345560	jp8ellin54g		1*	54 Mbps	(Fake)	AP	WEP	37	-34	-100	66	
00022DA709EE	a709ee		1	11 Mbps	Proxim...	AP		24	-76	-100	24	
000F66B0B8FD	kaming88		6	54 Mbps	Linksys	AP	WEP		-72	-100	28	

Figure 1.5.15

NetStumbler can easily detect which wireless access points are running without encryption in your neighbourhood. From the above figure, you can easily see that in our neighbourhood three out of four active access points are running in insecure mode. (Actually, only ours is running in secure mode.) The owners of these access points are virtually inviting hackers to invade their networks and computers!

You have successfully finished this lab.

## Summary

In this lab, you learned how to use a packet sniffer such as Wireshark to perform network traffic analysis. You also learned that information transmitted in unencrypted packets can be easily revealed by packet sniffing. Finally, you used a wireless networking tool, NetStumbler, to detect wireless networks running in insecure mode in your neighbourhood. Such networks are very vulnerable to packet sniffing. After finishing this lab, you should be more familiar with the important TCP/IP application protocols such as DNS, HTTP and POP3. In addition, we hope you have begun to appreciate the importance of network security, especially in a wireless network environment.

## Further reading and useful resources

- 'IEEE 802.11' — [http://en.wikipedia.org/wiki/WiFi%2C\\_802.11](http://en.wikipedia.org/wiki/WiFi%2C_802.11) (for more details about IEEE 802.11 standards)

## Questions and exercises

- 1 Name five popular TCP/IP application protocols that are vulnerable to packet sniffing.
- 2 Give a display filter that will show only HTTP POST packets.
- 3 Explain why packet sniffing is easier on a network that is connected by a hub than a switch.
- 4 If you are running a wireless LAN at home, investigate whether your access point is configured to run in a security mode. What are the available security modes on your access point?