# Lab-9-1: Network Traffic

In today’s lab, we are capturing and analysing network traffic. Specifically, we will generate some interesting network traffic and collect it using the Wireshark tool. We will analyse the collected data, and extract interesting information from it. In the next exercise, we will perform an analysis of previously collected network traffic and attempt to aid our client to find information about a suspicious employee.

**Exercise-9-1-1: Capturing and Analysing Network Traffic**

Today we will use Wireshark to capture network traffic entering and leaving our system. I hope that you remember some basic information about Wireshark from previous classes. If not, here are some useful links:

* [General introduction to Wireshark and what it can do](https://www.wireshark.org/docs/wsug_html_chunked/ChapterIntroduction.html)
* [Official Wireshark Wiki](https://wiki.wireshark.org/)
* [Wireshark Tutorial Video for Beginners](https://www.youtube.com/watch?v=6LGw31TsP6E&list=PLBf0hzazHTGPgyxeEj_9LBHiqjtNEjsgt&index=2)
* [A good summary by How-To-Geek](https://www.howtogeek.com/104278/how-to-use-wireshark-to-capture-filter-and-inspect-packets/) (but uses old Wireshark interface)

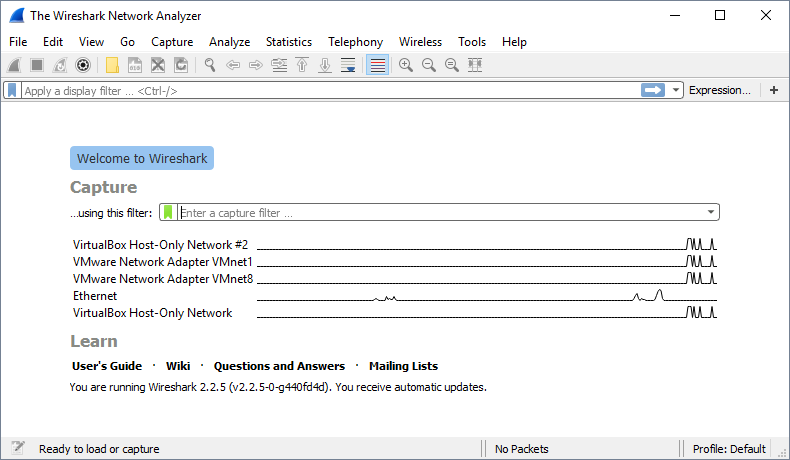
First, we need to get comfortable with the Wireshark interface. To achieve this we should start up Wireshark and collect some network traffic.

**NOTE: We are using our Windows 10 system today!**

Make sure you are logged in on your Otago Polytech Windows 10 system (not any virtual machines today). Start Wireshark. There are a variety of ways to accomplish this:

* Windows 10 Desktop > Locate “Wireshark” > Double-click icon to start
* Open Start Menu > Type “Wireshark” to search > Left-click icon to start

Once started, the default Wireshark interface will appear. The screenshot below shows an example of the Wireshark interface at startup.

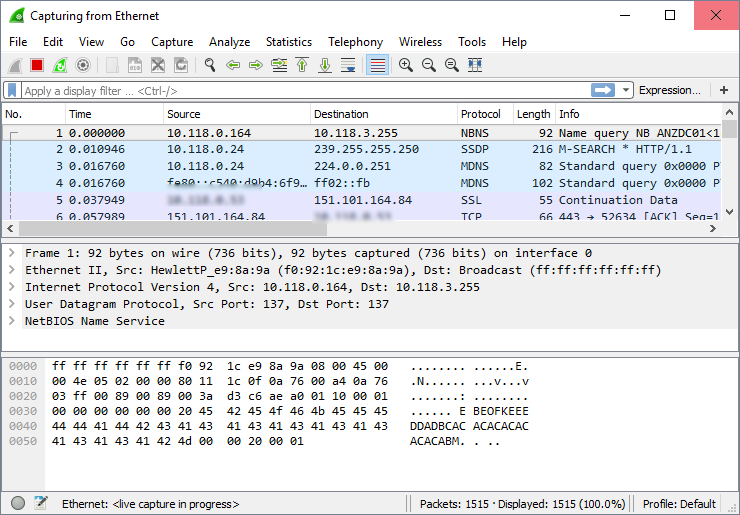


NOTE: The Wireshark interface changed dramatically in late 2015 when version 2.0 was released. Many tutorials use the old interface and it may be difficult following those tutorials.

Your interface may look a little different depending on the computer you are using. To start we need to specify the network interface we want to collect traffic from. In this exercise we are going to collect traffic on the “Ethernet” interface.

Double-click the “Ethernet” interface to start collecting network traffic.

Wireshark will start collecting traffic immediately. Your Wireshark interface should now look something like the screenshot below:



Refer back to the lecture slides for more information on each separate pane (window) within the Wireshark interface. Spend some time having a look at each of the packets, try browsing to a specific website and look at the traffic generated.

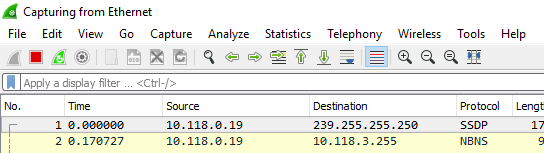
Stop the network traffic capture using the stop button (the red square button)

The image below shows the common buttons to 1) Start, 2) Stop, 3) Restart the process of collecting packets.

**STOP**

**RESTART**

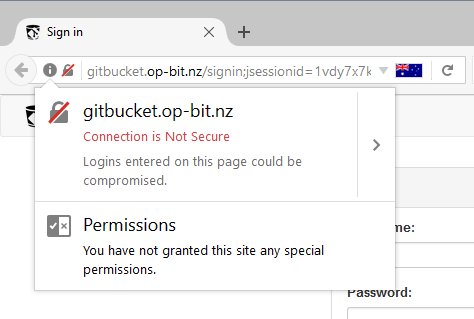
**START**



We have a very interesting target to test in today’s lab. In this course, we consistently use GitBucket to upload our graded lab exercise sheets. You may have noticed that the GitBucket system we use has a security issue (we have mentioned this briefly in class before). For reference, the URL of the GitBucket website we use is:

<http://gitbucket.op-bit.nz/>

When visiting the website (using the URL above) we get the following page in Mozilla Firefox. The Firefox browser is smart enough to know that this website may not be the most secure, and displays a padlock icon with a line through it. This means the website has no security – or more specifically – has not encryption to protect our username and password information! If you went to the GitBucket website, using Firefox, the following warning is displayed:



The first step to analysing anything from the GitBucket webpage is to determine the IP address of the server. This information will help us “filter” network traffic that is directly related to the website. There are two methods to find the server IP address:

1. Use the Windows Command Prompt (CMD):
   1. Open a Windows Command Prompt
   2. Open Start Menu > Type “cmd” > Double-click the “Command Prompt” icon
   3. Use the nslookup tool to determine the server IP from the domain name:
   4. nslookup gitbucket.op-bit.nz
2. Use an online nslookup server:
   1. Open a web browser
   2. Go to the following web site: <http://ping.eu/nslookup/>
   3. In the IP address or hostname box enter:
   4. gitbucket.op-bit.nz

**Q1.** What is the IP address of the GitBucket web server?

54.252.149.203

Now we are going to “sniff” the network traffic and log into the GitBucket website.

Navigate back to you previously opened Wireshark window. We need to start another traffic collection session. The following list of actions will help us collect the required traffic.

1. Use the “stop” button (the red square) to stop collecting network traffic (if you didn’t stop the network traffic collection previously).
2. Then use the “start” button (the blue fin) to start collecting traffic again. If you are prompted to save you previous captured packets, select > “Continue without Saving”.
3. Open a web browser – any will do (Firefox, Chrome or Internet Explorer)
4. Navigate to the GitBucket website: gitbucket.op-bit.nz
5. Enter your username and password
6. Switch back to Wireshark and hit the stop button (the red square)
7. Excellent! We have collected our network traffic

Now we have captured some network traffic, we shall have a look at the overall statistics of the data we collected. Navigate to the “Statistics” menu in the application and then select “Capture file properties”. Alternatively, you could use the shortcut Ctrl + Shift + Alt + C. Using the information provided answer the following question:

**Q2.** Document the following properties: 1) Elapsed time of the packet capture; 2) Total number of packets collected; 3) The average number of Packets per Second (or PPS).

1: 00:00:24

2: 598

3: 24.6

We talked about Wireshark filters in the lecture today. We will use these to help reduce the number of packets that require manual review.

First, we can filter all the packets that contain the IP address of the server (the IP address required is your answer to question one). The following filter can be used:

ip.addr == <ip address of server>

For example:

ip.addr == 192.168.19.19

Press “Enter” to apply the filter. If the filter is valid, the background of the filter pane should change to a green colour. Now, we are going to try another filter. This time we are interested in network traffic that contains HTTP information to display web pages. Specifically, we are interesting the HTTP GET requests. These are used to ask the server for a resource; for example, requesting the web server for an image to load on a web page. Try the following filer:

http.request.method == "GET"

If there is still a large amount of network packets (more than 10-20), try adding two filters together. This can be achieved by using a double ampersand (&&):

http.request.method == "GET" && ip.addr == <ip address of server>

**Q3.** Try to find an interesting HTTP GET request (e.g., downloading of an image). Hint: Look for additional information in the “Info” column, or inspect the packet contents. Describe what the GET request is fetching.

GET /assets/common/images/gitbucket.png HTTP/1.1\r\n

Lastly, we are going to look for an interesting security issue. Remember when we logged into the GitBucket website, we entered a username and password. However, there is a security issue with the website which we can leverage to get information. Try the following Wireshark filter:

http.request.method == "POST"

If there are more than two packets, try to add another filter specifying the IP address of the server (from question 1). Perform further analysis of the packets found. Is there any interesting information available? HINT: Try to find the “HTML Form URL Encoded” field.

**Q4.** What information did you find?

HTML Form URL Encoded: application/x-www-form-urlencoded contains my Username and Pasword

**Q5.** What security issue caused this problem?

No encryption. Plain text.

No HTTPS, not secure.

**Exercise-9-1-2: Ann Dercover is a Secret Agent**

Riveting Backstory: You have been hired by Anarchy-R-Us, Inc. as a security expert to investigate suspicious activity of one of their employee’s, Ann Dercover. The company attempted to prosecute Ann – a known undercover secret agent who is actually working for a competitor of the company. Unfortunately, Ann was released on bail and has skipped town.

Luckily, the security team has been monitoring Ann’s activity and network traffic for some time now. They have captured network traffic from Ann’s laptop that seems to contain two emails. She has been in contact with her secret lover, Mr X, before she left….

NOTE: To perform the tasks for this lab we have been provided a previously collected packet capture file (evidence02.pcap) from the security team. It is available on the I:\ drive.

As the security expert tasked with this investigation, your mission is to discover the following information.

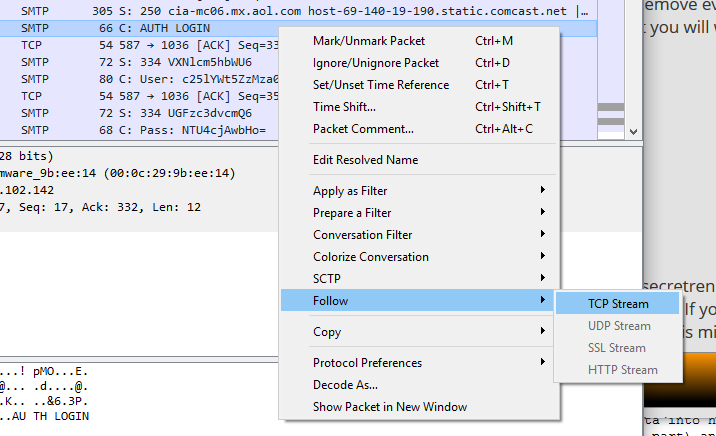
1. What is Ann’s email address?
2. What is Ann’s email password?
3. What is Ann’s secret lovers email address?
4. What two items did Ann tell Mr X to bring?
5. Bonus: Extract the email attachment and determine rendezvous point

To help you out, we will work through the exercise together. Firstly, we know Ann is using email to communicate. If we have a quick look at the “Protocol” column of the packet capture there seems to be many packets using the SMTP or the Simple Mail Transfer Protocol. So we will use to power of Wireshark filters to only show SMTP…. Try the following filter:

smtp

Good job! Only SMTP network packets are displayed. Now we can use a very powerful feature of Wireshark called reassembling TCP streams. This means that we can group all associated packets together; for example, all packets associated with a single email.

In SMTP, the first packet in a conversation will usually start with “AUTH LOGIN” – which occurs when a user logs in to their email account. We can right click this first packet, select “Follow” and then “TCP Stream”. Have a look at the image below for an example:



After running this function, a new window should appear containing information about the email. Wireshark amalgamates this information by combining multiple packets into a single interface.

Using the information presented in this new reassembled TCP stream, we can start answering some of the questions. HINT: Whenever we select a line in the new window, the packet, which contains that information, is highlighted in Wireshark.

**Q6.** What is Ann’s email address?

sneakyg33k@aol.com

**Q7.** What is Ann’s email address password? HINT: It is common in SMTP to encode information in Base64. You can easily decode a string in Base64 using an online tool such as <https://www.base64decode.org/>

558r00lz

There are two emails contained in this packet capture file. Find the second email by examining the “Info” column and searching for the second appearance of “AUTH LOGIN”. You may see an email close by that has the address of Ann’s secret lover: Mr X. We can reassemble this email by right-clicking one of the packets, selecting “Follow” then selecting “TCP stream”.

**Q8.** What is Ann’s secret lovers email address?

mistersecretx@aol.com

**Q9.** What two items did Ann tell Mr X to bring?

Fake passport and a bathing suit

**BONUS Q10.** Extract the email attachment and determine rendezvous point. Provide a map or an address.

