# CIT 285 - Lab #1: Packet Sniffing

## Introduction

The best way to understand any communication over a network is to watch the packets transferred during the communication. A packet sniffer is a tool for performing that task. As the name suggests, a packet sniffer watches the packets being sent or received by your computer. It does so by creating copies of the packets as the network interface card receives them. Your network applications still receive the packets as they normally would, so the packet sniffer does not interfere with system operation. Once you have a copy of the packets from the sniffer, you can analyze them.

However, the binary data exchanged over the network is usually opaque and difficult for humans to read. To solve that problem, we use a protocol analyzer, a tool that decodes the binary packet data into a humanly readable format. In this lab, we will use Wireshark, which calls the dumpcap tool to sniff packets, then provides network protocol analysis via a graphical user interface. The GUI allows the administrator to see an overview of packets, filter packets by protocol or any packet header, and to show the contents of a single packet layer by layer through the TCP/IP protocol model.

There is also a command line version of Wireshark called tshark that provides the same capabilities in a text interface, which is necessary when performing analysis on a server without a GUI or when performing live packet analysis on a high throughput network connection, where the GUI cannot keep up with network throughput.

**For this lab, you will need a client and a server VM. We will use the following two VMs:**

1. Kali Linux
2. Metasploitable 2

We will run Wireshark on the Kali Linux VM.

**References**

1. Wireshark documentation, including user manual and videos, https://wireshark.org/docs/
2. http://www.howtogeek.com/104278/how-to-use-wireshark-to-capture-filter-and-inspect-packets/

## 1: Basic Packet Sniffing with Wireshark

Boot the Kali Linux VM. Do NOT boot the Metasploitable VM yet.

Start Wireshark from the terminal with the following command:

**# wireshark &**

Click past any warning dialogs. We do not need the features that Wireshark is disabling to protect the system since we are running Wireshark as root. Maximize the Wireshark window.

Before capturing packets, you must select a network interface from the interface list in the Wireshark GUI. **Double-click on eth0**, the virtual machine's only Ethernet interface. (*If no interfaces are listed, there has been a mistake made in the setup lab completed previously*.)

In a second terminal, ping google with the command below:

**# ping -c 3 www.google.com**

Click the red square to Stop the current capture session when the ping complete.

Packets are listed one per line in the top pane of the Wireshark GUI.

* The No. column shows the packet number
* Time shows time elapsed since packet sniffing began
* Source and Destination show the source and destination MAC or IP address
  + Depending on whether the protocol is working at the Data Link or Network layer (or higher)
* The Protocol column shows the protocol of each packet
* Length shows packet size
* Info provides a brief description of the meaning of the packet.

1.1: Which protocols do you see in your captured packets? List or screenshot them below.

**Graphical user interface, application

Description automatically generated**

1.2: Which **IP addresses** (source and destination) do you see in the captured packets?

**For source IP addresses I see all of 10.2.240.202, 10.11.0.51, 172.217.4.196, Vmware\_b9:d2:84,8.8.8.8**

**For destination IP addresses I see all of 10.2.240.202, 10.11.0.51, 172.217.4.196, 8.8.8.8 and Broadcast**

1.3: Some of the IP addresses you see may begin with 224.

**What is special about such addresses?** You may need to look at your notes from your networking class or lookup such addresses online to refresh your memory.

**This address is a multicast address meaning that this address can be subscribed to multiple network interfaces.**

1.4: In the filter textbox type ‘arp’ (without the quotes) and hit Enter to tell Wireshark to show only ARP packets.

**Which addresses are being “looked up” in the ARP packets?** You may need to scroll the packet window bar to the right to see all information.

**The source address which is the vmware are 00:50:56:b9:xx:xx and the destination address or the broadcast is the ff;ff;ff;ff;ff;ff**

In the **Capture** menu, click on **Restart or Start** to clear the list of packets and begin capturing again. You will not need to save the capture.

While Wireshark is running, bootup your Metasploit VM and watch Wireshark as the VM loads. Watch for packets from the Metasploitable VM’s IP or MAC address in Wireshark as Metasploitable2 boots. If you do not see any packets, be sure to remove the ‘arp’ filter.

When a machine boots, it will send packets using one or more network protocols to provide information about itself to servers or to obtain information. DHCP is an example of a protocol used at boot time, but we should not see DHCP packets since Metasploitable2 has a static IP address.

**1.5: Provide a screenshot of the protocols received and briefly describe the information revealed or obtained by one or two of the packets during Metasploits bootup**.

Graphical user interface

Description automatically generated

**So when looking at the data packs there is a bunch of new tabs and information such as NetBIOS datagram service, SMB, SMB MailSlot Protocol, and Microsoft Windows Browser Protocol. We can see under the Microsoft Windows Browser Protocol the major and minor version for the browser protocol. We can also see sizes in the SMB.**

## 2: Sniffing an HTTP Connection

In the **Capture** menu of Wireshark, click on Restart**. Enter “http” as your filter in the Filter toolbar**. Start a web browser on the Kali VM and type [www.nku.edu](http://www.nku.edu/) in the location bar to load the NKU home page, creating a number of HTTP requests. Once the home page is loaded, stop capturing packets.

To limit Wireshark’s display to only packets sent or received by your Kali VM, enter the following expression, where **KALI\_IP** is your Kali IP address.

**ip.addr == KALI\_IP**

If needed, you can further limit the display of packets by protocol by adding a logical AND operation followed by a protocol specifier, e.g. “&& http”, after the IP address specifier.

2.1 How many **HTTP requests** (not total packets) do you observe from starting your browser? What are the requested URLs? (list no more than 5).

**From starting the browser you get one http request and one reply, and then repeated once more at the end. The urls I see requested are such ones like** [**http://www.nku.edu**](http://www.nku.edu) **and http://detectportal.firefox.com/success.txt.**

2.2: Select each **HTTP request** in the packet window expand the ‘w plus sign in the packet details window to locate the request for [www.nku.edu](http://www.nku.edu).

* **Were you able to see the actual GET request from your Kali VM to** [**www.nku.edu**](http://www.nku.edu)**?**
* **What version of your web browser is running on your system?**
* **What are the compatible browser versions?**

**Yes you are actual able to see the GET request from your Kali VM to** [**www.nku.edu**](http://www.nku.edu)**. The version of the web browser we are running is Apache/2.4.6. Mozilla, Firefox, Gecko.**

## 3: Sniffing an HTTPS connection

Next, we will examine an HTTPS session, in which the HTTP packets are encrypted with SSL/TLS. Restart Wireshark and in a new browser window, enter the URL <https://www.google.com/> in the location bar. Let the page load and stop capturing packets once it does.

Apply a filter of ‘**tls**’ to reduce the number of packets. Note:

* *To answer the following questions, click on the packet of interest, then go to the Packet Details pane and click on Transport Layer Security to view its SSL/TLS information.*
* *You will need to drill down in the headers to find the needed content*

3.1 Examine the first **Client Hello** packet **sent from Kali to Google**.

* What version of TLS is requested by this packet?
* What version of TLS is current? Check Wikipedia if you do not know.

TLS 1.2 and the current version is TLS 1.3

3.2: Examine the first **Server Hello** packet **sent by Google to Kali**. Expand the Transport Security Layer headers until you locate the ‘CipherSuite’ line. Based on that text string, identify the

* Public key algorithm
* Symmetric algorithm
* Hash family that this TLS session will be using
* *If you see multiple cipher suites, you are most likely still looking at the Client Hello packet. The Server Hello packet should only have one cipher suite.*
* *For additional help, use the following links:*
  + [Wikipedia – Cipher Suites](https://en.wikipedia.org/wiki/Cipher_suite)
  + [*SSL Store – Cipher Suites and Algorithms*](https://www.thesslstore.com/blog/cipher-suites-algorithms-security-settings/)

**Public key algorthim poly, symmetric algorithm is ECDSA. The hash fanily that the TLS session uses is the RSA.**

3.3: Find the first ‘**Application Data**’ packet after the Client/Server Hello exchange. Keep the Transport Layer Security header expanded.

* What information is provided in the Packet details window about the data captured in this packet?
* Can you view this information? Why or Why not?
* Does this change your answer to question 2.2 any?

**Opaque type, Version, length, and encrypted application data. We cannot access this information; we do not have the permission or access to do so because it has to deal with the other application. Yes it makes me want to rethink my answer on 2.2.**

## Submission:

Upload a completed copy of this document to Canvas by the due date.