**Laboratory 1 – Introduction to Wireshark**

Due Date: 7th Feb, Sunday.

**1. Overview**

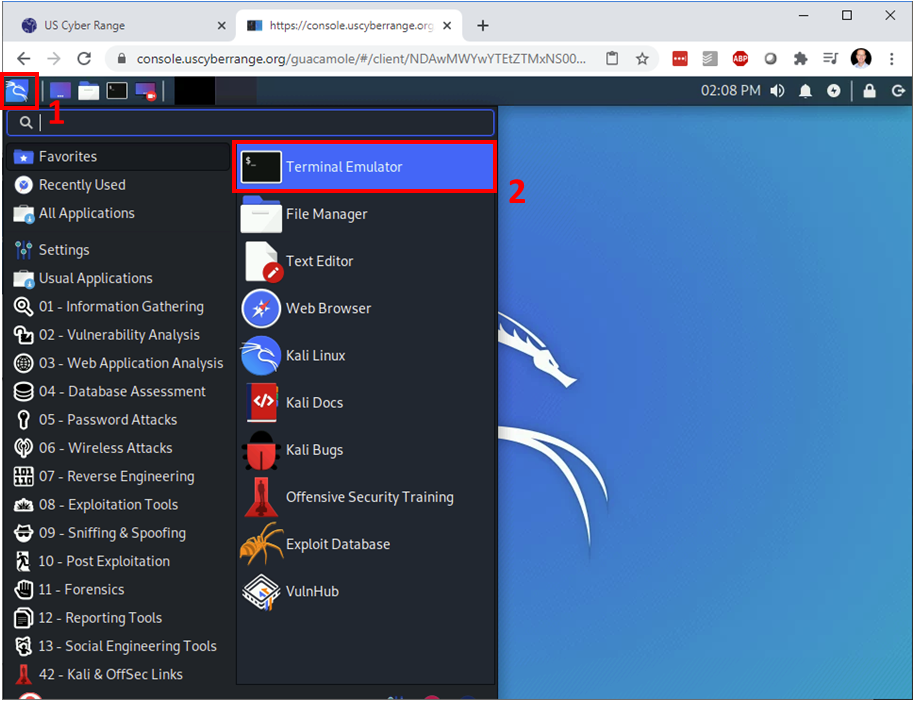
In this lesson, the student will be introduced to Wireshark, a very useful tool that covers a very important network forensics concept – reading and understanding networking traffic. Wireshark (software known as a packet analyzer) allows you to view pieces of data (called packets) in real-time as they go in and out of a system and can be saved as packet capture (pcap or cap) files. In this exercise, the student will be analyzing packet capture files as well as capturing live network traffic in real-time.

**2. Resources required**

The lab will be completed using the Cyber Range “Cyber Basics (2020.09)” multi-Virtual Machine environment.

**3. Initial Setup**

This exercise requires a Kali Linux Virtual Machine in the Cyber Range (or in a VM on a laptop will work too). If using Cyber Range account, login to your Kali Linux virtual machine (username/password = student / student). You can launch a terminal window by accessing it through the **Applications** menu at the top left and selecting **Terminal Emulator.**



**4. Tasks**

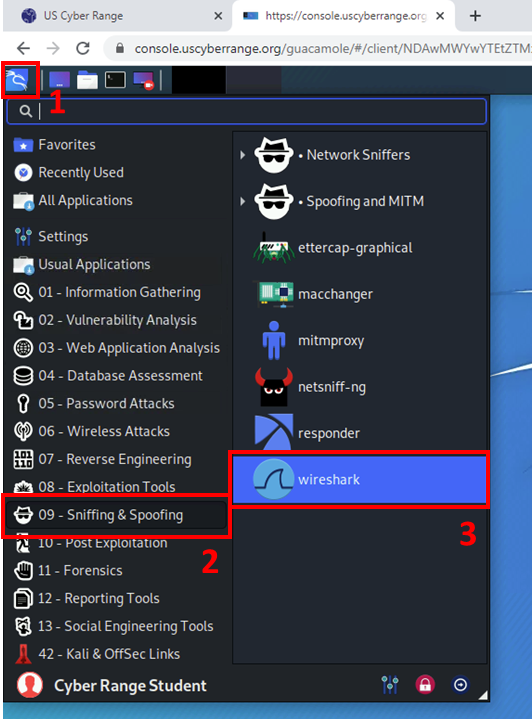
**Task 4.1: Getting familiar with Wireshark by observing live capture**

**What is Wireshark and why should you care?** Wireshark is a powerful analysis tool that allows you to not only capture network and device packets, but to analyze them too, a packet is a fragment of data that is sent over a network from one machine to another. This data usually includes a *source port*, *source IP* address, *destination por*t, *destination IP*, and other data that we will see in this lab.

Wireshark allows a user to analyze the traffic traveling in and out of the machine, which can serve many uses. These include, but are not limited to:

* Troubleshooting network connections.
* Filtering data between two hosts to see a single network “conversation.”
* Comparing all “conversations” to discover bad actors or “bandwidth hogs.”  
  ***NOTE:*** *The endpoint using the most bandwidth is known as the “top talker.”*
* Filtering captured data to analyze specific protocols and ports being used.
* Analyzing specific statistics about the traffic coming in and out of the system.

**How do we use Wireshark?** Let’s take a look at this application. Open Wireshark either by clicking on the Applications menu in the upper-left corner of the Kali desktop, then click on Sniffing&Spoofing, then Wireshark. You will be required to enter your sudo password – enter “student”. ***NOTE:*** *Running Wireshark using* ***sudo*** *runs it with root-system privileges and is needed for some operations (like reading live network traffic).*



Taking a closer look at the Wireshark user interface, let's explore how it's laid out.

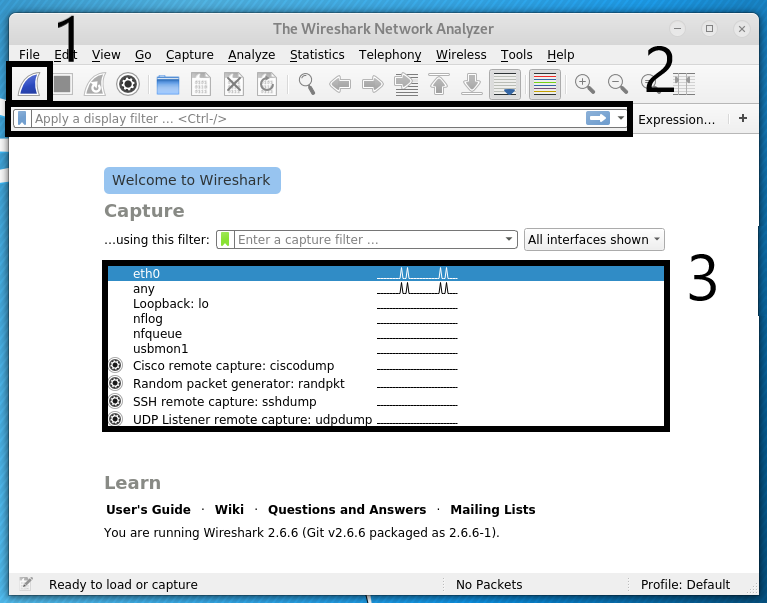


Figure 3 - The Wireshark opening user interface.

From Figure 3 above, let’s breakdown some of the important Wireshark features outlined by the three numbered boxes:

1. **Start Capturing Packets:** This is the button to start a live packet capture. This will capture network traffic going in and out of your system in real-time. We will demo this later in the lab.
2. **Filter String Field:** This field allows the user to apply filters to the traffic captured. This can be done by certain text, a protocol, a port, etc.. We will look at applying filters later.
3. **Live Packet Devices:** This is a list of the network interfaces from which we can capture traffic.

So, if we wish to look traffic for the eth0 network interface, highlight the “eth0” interfaces (shown above *highlighted in blue in Box #3*) and press the start button (Box #1) or double-click the “eth0.” After doing so, the live packet capture will record all packet data traversing the host running Wireshark.

***WARNING:*** *Capturing or recording live packet traffic takes up a very large amount of memory. Be mindful of this and never just leave a Wireshark capture running. Doing so for just a few minutes, can fill up Gigabytes of system memory or fill up your disk when trying to save. Both of these situations can make your system unstable or crash Wireshark. Stop Wireshark by hitting the button in Box #1 as shown below in Figure 4.*

Now we will take a look at some important packet data capture Wireshark features outlined by the four numbered boxes below in Figure 4.

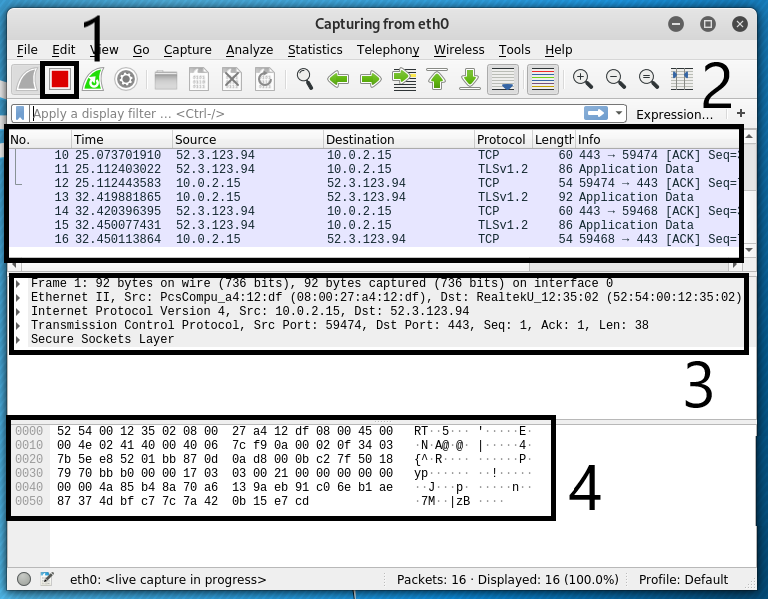


Figure 4 The three main capture inspection frames in Wireshark

1. **Stop Capture Button:** This button stops the current capture. Once you click this, you can analyze the data and then save it as a .pcap file (a file containing captured packet data) for further analysis or exporting.

***NOTE:*** *Once you capture data, you can save it by simply opening* ***File / Save*** *and giving it a file name. By default, it will save the newer file format called .pcapng, but you can also save in the more classic .pcap format as well as a couple dozen other network capture formats.*

1. **Packet List Pane:** This displays the captured data packets. It shows the Source IP, Destination IP, Protocol, etc. of each packet captured.
2. **Packet Details Pane:** This displays the data structure contained within the selected packet from the Packet List Pane (in Box #2). In the details pane, you can clearly see the Layer-1 through Layer-3 TCP/IP encapsulated data structures. From the top, the Layer-1 frame & Ethernet data, wrapped in a Layer-2 IP datagram, in turn wrapped in a Layer-3 TCP transport session or connection.
3. **Packet Bytes Pane:** This displays the raw data of the highlighted packet (in Box #2) in its most basic or “canonical” hexadecimal + ASCII formats — the lowest level, most basic, binary data, represented in both hex (machine) and ASCII (human) readable formats side-by-side.

Now that we understand how Wireshark is used to capture data and then export it to a file as pcap file, let’s take a look at a few pcap files, interpret the data in them, and see what we can learn!

**Task 4.2: Analyzing pcap files:**

Let's first download several .pcap files for analysis from the Wireshark website.

At the terminal, type the following:

$ **firefox https://gitlab.com/wireshark/wireshark/-/wikis/samplecaptures**

Scroll down to find and download (Save File) the following files:

Under heading of General / Unsorted

* dhcp.pcap
* dns.cap

Under heading of Viruses and worms

* slammer.pcap

Close any previously opened Wireshark sessions, and then at the terminal, input the following commands:

$ **cd ~/Downloads**

$ **ls -lah \*cap**

-rw-r--r-- 1 student student 1.4K Jan 30 21:46 dhcp.pcap

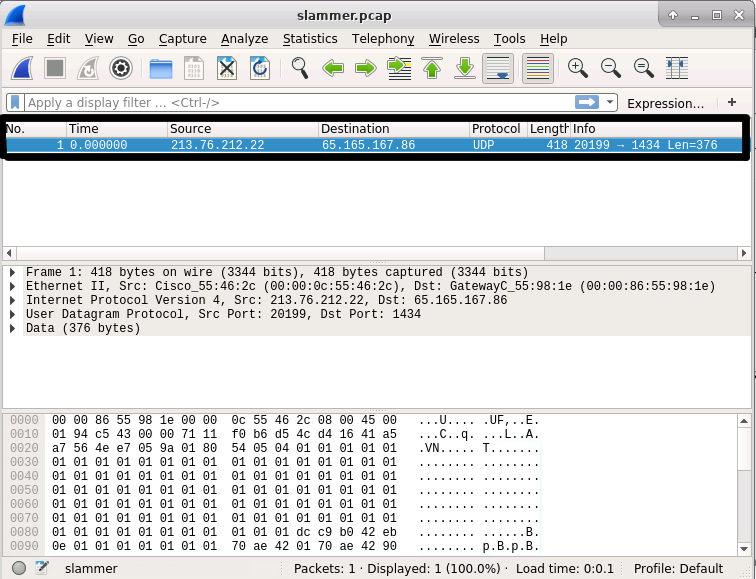
-rw-r--r-- 1 student student 4.3K Jan 30 21:47 dns.cap

-rw-r--r-- 1 student student 458 Jan 30 21:47 slammer.pcap

$ **wireshark slammer.pcap**

This **slammer.pcap** packet capture we're opening contains a single packet; it's used for simplicity’s sake.

Looking at the top packet in the Packet List Pane in Figure 5, let's see what information we can gather.

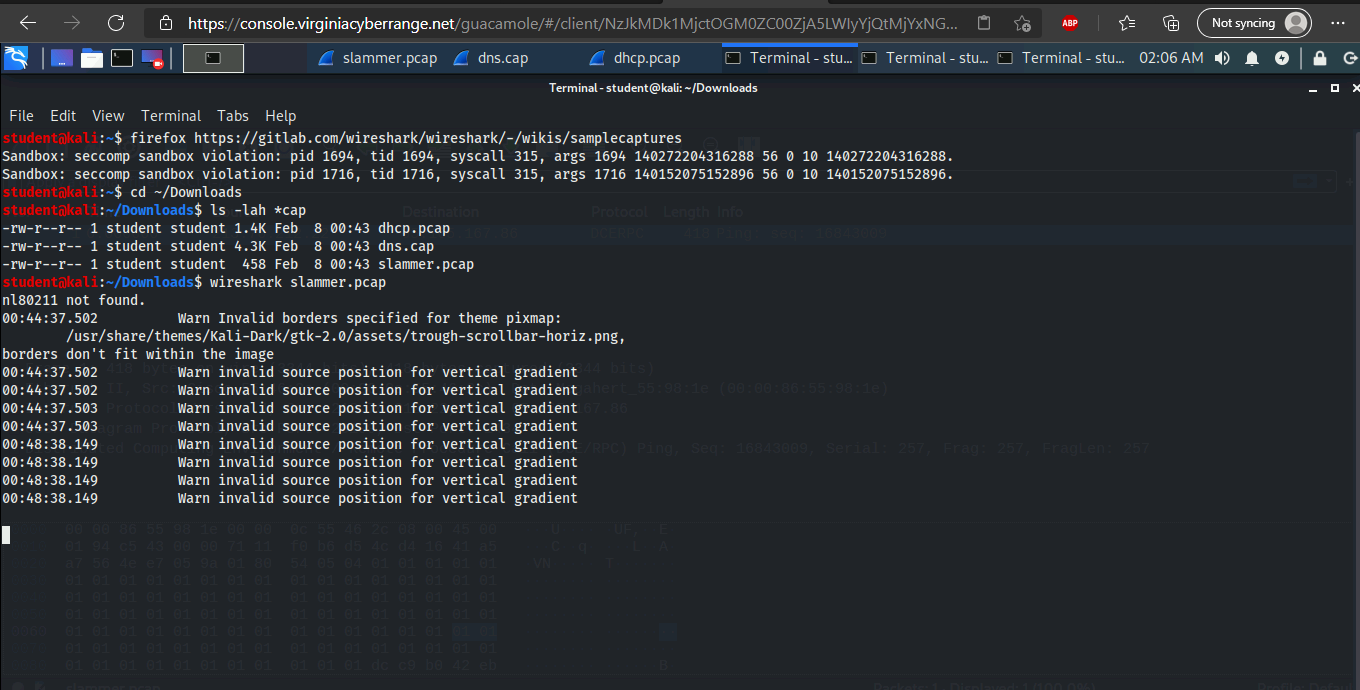
Figure 5 5: Inspecting a single packet from the slammer pcap file

If you view the top window, this pane lists all of the packets in the capture (in this case, a single packet), as well as other basic information about the packets. The table below explains what each field means.

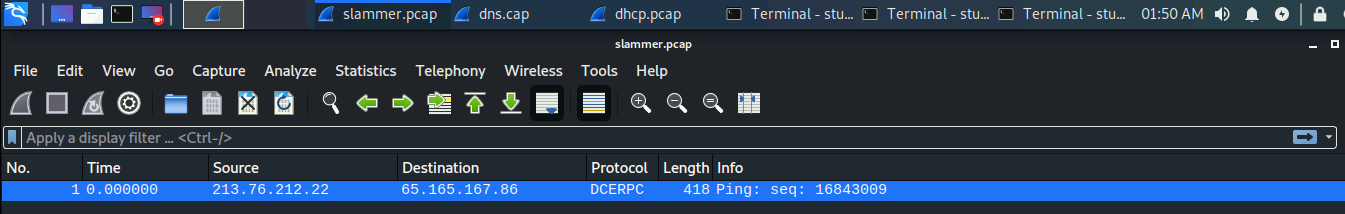
|  |  |
| --- | --- |
| **Field Name** | **Description** |
| No. | The packet number; this goes incrementally in order by which packets reach the machine or leave the machine first. |
| Time | This is the time from the first packet capture to the selected one; if it is the first packet, the time is 0.0. |
| Source | The IP address of the machine the packet originated from. |
| Destination | The IP address of the intended recipient of the packet. |
| Protocol | The networking protocol used to send this packet. In Wireshark, if we desire we can filter captured data based on specific protocols. Recall from lesson 5 (Intro to Networking) where we introduced and elaborated on TCP and UDP. Here, we are seeing a real-world application of what we learned in the “Intro to Networking Lab” where we had discussed protocols and their uses. Specifically, this data is Network Layer traffic and we can see this packet is using the UDP protocol. |
| Length | The data length/size of the packet. |
| Info | Details about the packet; may be helpful or contain little value. |

The Packet Details pane allows you to drill down into the details of the packet highlighted up in the Packet List Pane.

In the packet in Figure 5 from the file **slammer.pcap**, use the Packet Details Pane to drill down into the details of this single highlighted packet and answer the questions below.

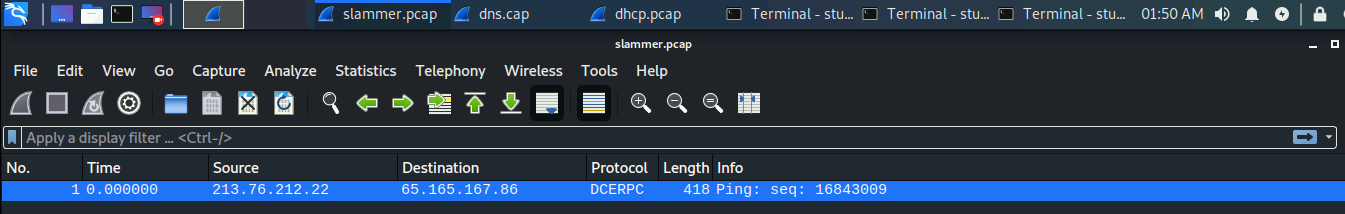


**Question 4.2.1: What is the *source* *port* of the packet in question?**



**213.76.212.22**

**Question 4.2.2: What is the *destination port* of the packet in question?**



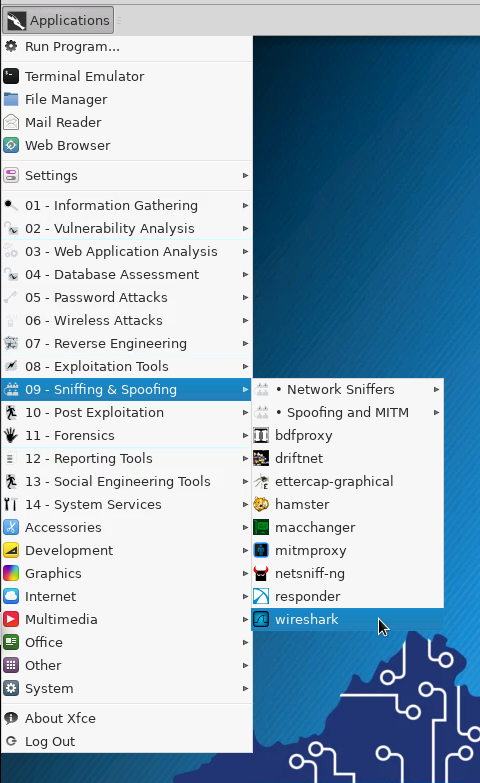
**65.165.167.86**

**Task 4.3: Analyzing DNS Packets**

DNS (Domain Name System) is a network service that runs over TCP/UDP port 53 and translates human readable domain names ([www.example.com](http://www.example.com/)) into IP Addresses (10.234.123.45). For example, when you enter www.facebook.com into your web browser, your computer or network provider's DNS service translates this human readable FQDN (fully qualified domain name) into an IP Address so the browser can use it to retrieve or “*GET*” the website.

Why is name-to-IP address-resolution even needed? As you know, all data within computers (e.g., memory addresses, binary machine code, IP addresses, etc.) are all ultimately interpreted using numbers and binary data. These forms of data are not as easily interpreted by humans as are words. It is easier for us to remember “kali.example.com” rather than an IP address. Imagine if every website you wish to visit, you would have to remember some IP address such as 10.43.233.78. Instead, we remember domains and URLs, and DNS translates these into their respective IP addresses for the computer host to translate and interpret.

***DIG-IN:*** *To see how DNS works, open a terminal in your Kali Linux VM on the Cyber Range and look up the DNS record to translate google.com into its IP address by typing the DNS lookup command:* ***host google.com***

Figure 65

*This will resolve (or translate) the domain google.com to one or more IPv4 addresses (called A records), IPv6 addresses (called AAAA records), and also provide the known email servers that handle email traffic for that domain (called MX records). This type of DNS resolution happens every time you visit a website, click on a link, or send an email.*

Exit and close down any other running Wireshark instances and either start a new Wireshark instance from the Kali UI (upper left) by clicking on **Applications / 09 – Sniffing & Spoofing / Wireshark** (See Figure 6 on the left.). Once Wireshark starts, open the dns.cap file by selecting Wireshark's **File / Open Ctrl+O**.

If you prefer starting applications from the command line, instead of using the desktop UI, try opening the dns.cap like this:

$ **cd Downloads**

$ **wireshark dns.cap &**

***NOTE:*** *Ending a command with* ***&*** *puts the process in the background. This allows you to continue to use the terminal and your hands never have to leave the keyboard.*

Right away, you'll notice this DNS capture (dns.cap) has many more packets than the last one. See Figure 7 below. In the Packet List pane, select the 10th packet (No. 10) and explore some of the content in the Packet Details pane:

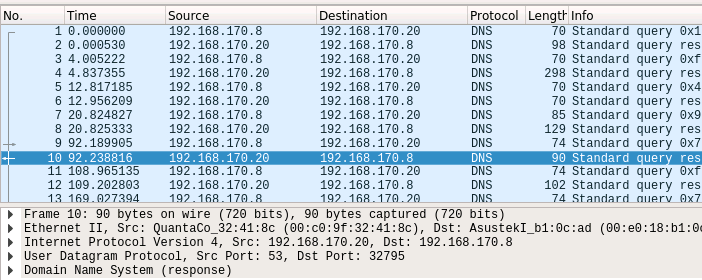
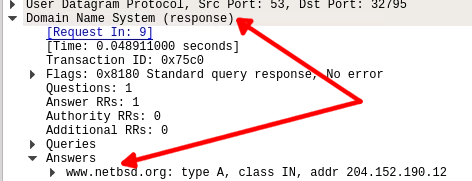


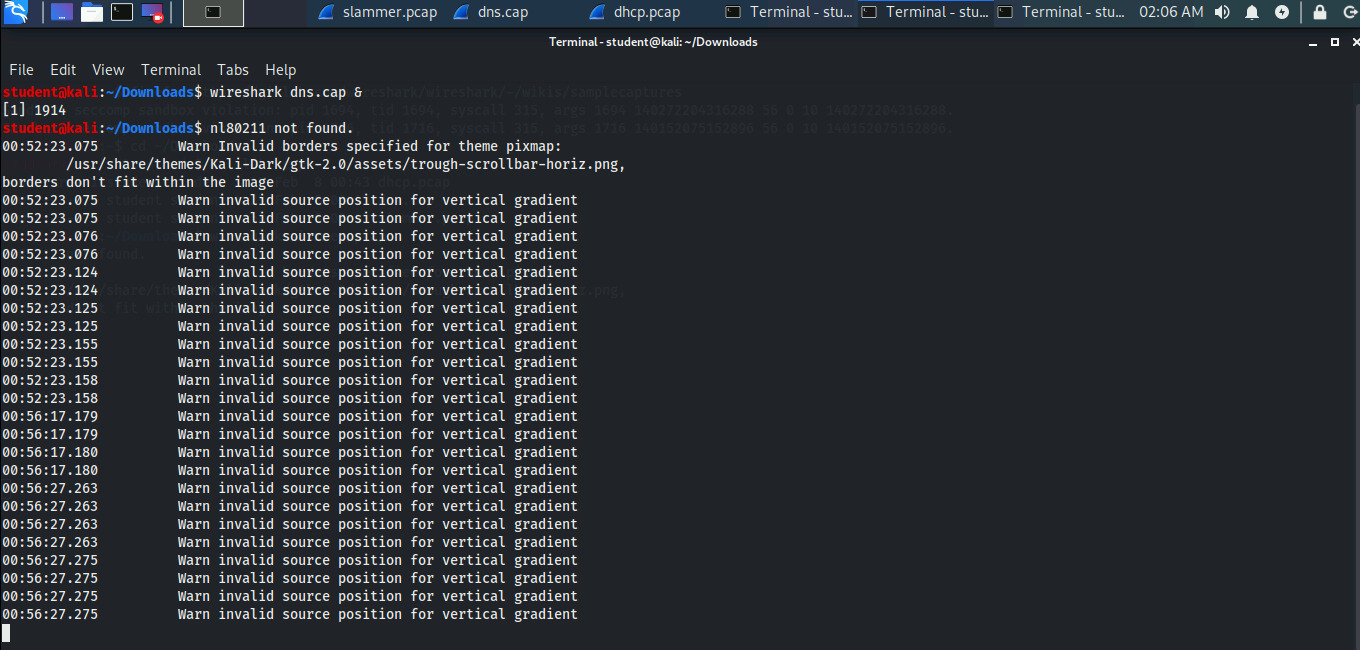
Figure 7 Exploring the details of packet #10

In the Packet Details pane, we are able to examine much more of the packet's internal details. Let's examine the DNS information for packet no. 10. See Figure 8 below.

Figure 8 Looking at the DNS response

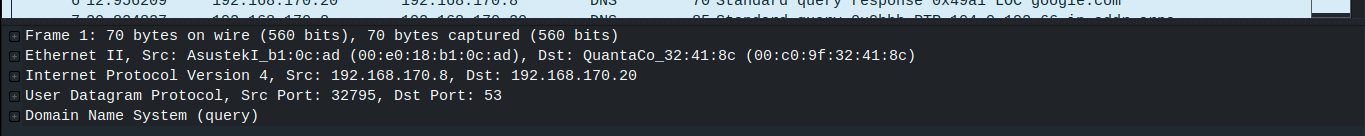


Expanding the **Domain Name System (response)** and then the **Answers** tabs, you will notice the IP response for the domain [www.netbsd.org](http://www.netbsd.org/). This means that a query was made for the domain [www.netbsd.org](http://www.netbsd.org/) and the IP address was successfully retrieved.

****

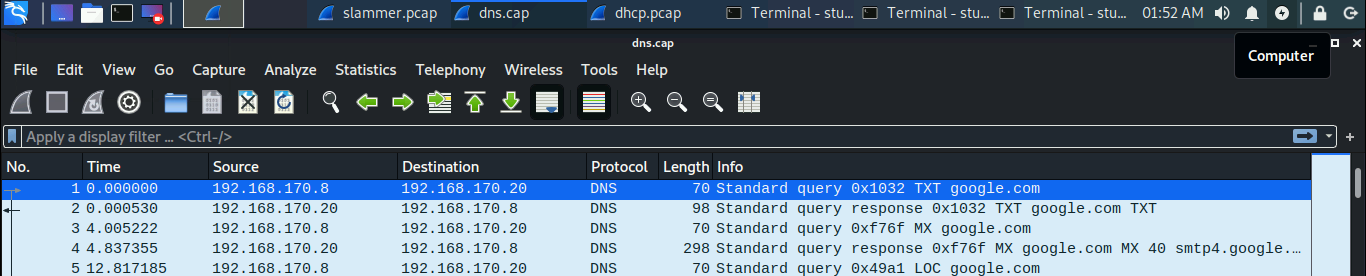
**Question 4.3.1: In Packet No. 28, what is the Source Port of the UDP data?**

**(Hint: What does UDP stand for?)**



**32795**

**Question 4.3.2: In Packet No. 5, what URL was queried for DNS information?**



**Standard query 0x49a1 LOC google.com**

**Task 4.4: Analyzing DHCP**

Previously, we discussed what IP addresses are and what role they play in a network. Now we are going to discuss how a device gets configured with an IP address on the network.

There are two methods for a system getting an IP address:

1. **Static IPs –** Some devices allow you to manually set an IP address in the network settings of the device. This requires a user to log into their device, configure the IP address to match the LAN settings, and save the IP address for future sessions. Static IP addresses are suitable for devices that you want to have constant (unchanging) IP settings such as a printer, a web server, or a file server. Generically speaking, static IP addresses are mostly used on servers and infrastructure devices that should not (or cannot) change IPs.
2. **Dynamic IPs –** Devices can also have IPs automatically assigned via *DHCP* (dynamic host configuration protocol), which automatically assigns IP addresses from dynamic IP address pools, pre-configured by the network administrators to allow devices (mostly human client hosts). DHCP-assigned IPs tend to be less prone to human error as they are automatically assigned when a device connects to a LAN (local area network). DHCP-based IP configurations are much more common on LANs than static IPs, and are used almost exclusively on WiFi networks.

Let's take a closer look at what DHCP traffic looks like on a local network.

Close down any Wireshark instances. Launch Wireshark and open the downloaded dhcp.pcap file by typing the following in a terminal:

$ **cd ~/Downloads**

$ **wireshark dhcp.pcap &**

***NOTE:*** *The* ***cd ~/*** *part of this command sets your working directory to your home path /home/$USER. So, by typing* ***cd ~/Downloads*** *in a terminal on the Cyber Range Kali VM, you will land in the /home/student/Downloads directory.*

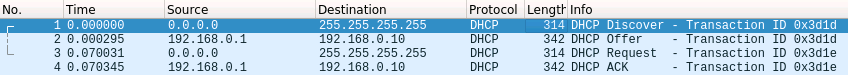
Looking at the four packets from this capture (dhcp.pcap) as shown in Figure 9, they clearly illustrate how DHCP works. When a device like your computer is connected to a network, it interacts with the DHCP server in the following way:

1. DHCP Discover – Client broadcasts a discovery request to all devices on the network in an

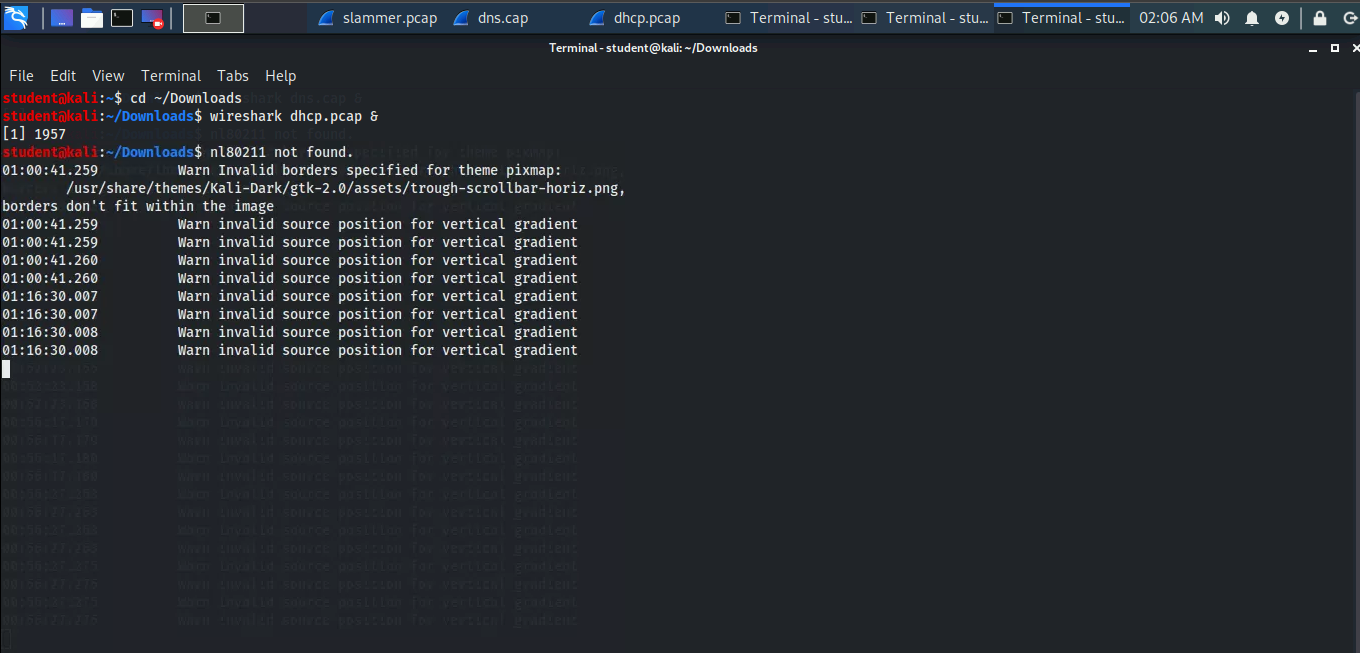
attempt to reach a DHCP server.

1. DHCP Offer – An IP lease is offered to the client by the DHCP server.
2. DHCP Request – The client then requests an IP lease from the server.
3. DHCP ACK(knowledge) – Lastly the server acknowledges an IP lease.

Now the device has been assigned a leased IP address.

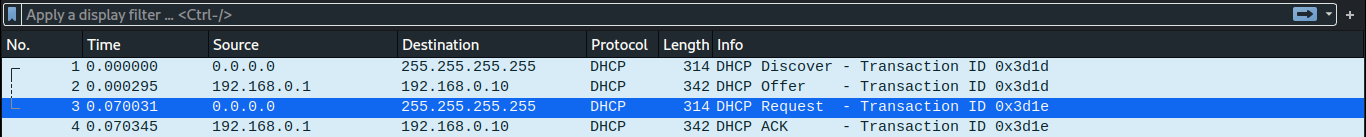
Figure 9 The DHCP client & server packet exchange showing how an IP address is assigned via DHCP. 

As you will see, the 4-step process that assigns the dynamic IP address to the device is listed in the four packets in Wireshark. Analyze the four packets in this packet capture and answer the questions below.

****

**Question 4.4.1: After looking at the packets, what are the 4 phases of a DHCP request?**

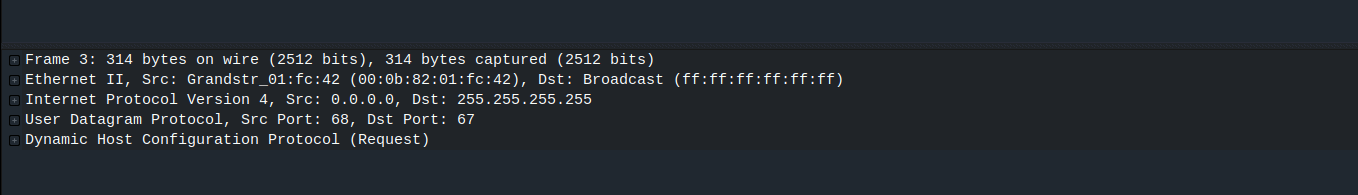
Discover, Offer, Request, ACK



**Question 4.4.2: Out of the two transport protocols (TCP or UDP), which is used for DHCP? Why do you think this is?**

TCP, because it is reliable as a transport, has flow and congestion control, and is connection oriented.

**Question 4.4.3: What incoming server port does DHCP rely on to receive requests? (Hint: Look at the Packet Details Pane.)**

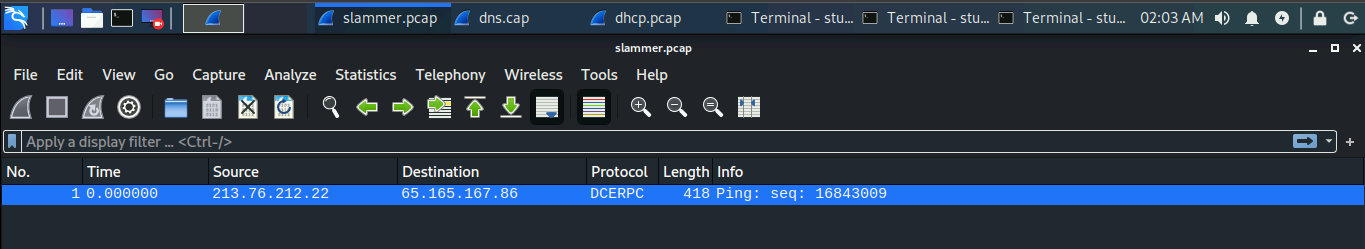


Internet Protocol Version 4, Src: 0.0.0.0, Dst: 255.255.255.255

**End of Lab Questions:**

1. From our slammer.pcap file, what is the destination IP of the packet?

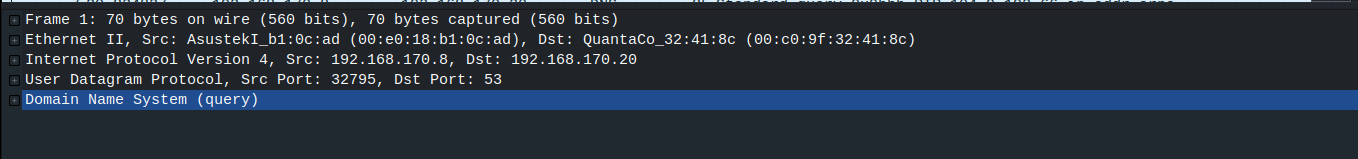
65.165.167.86



1. What is DNS?

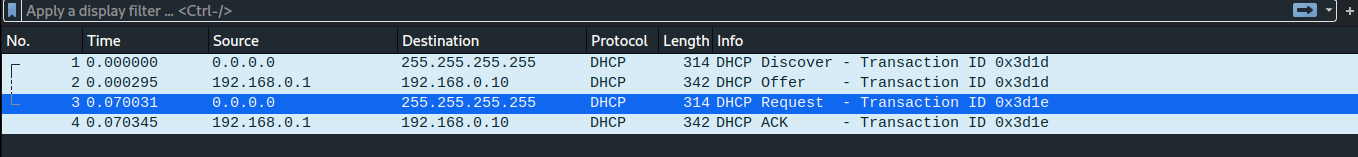
DNS is a network service that runs over TCP/UDP port 53 and translates human readable domain names into IP Addresses.

1. What port does DNS listen on?



User Datagram Protocol, Src Port: 32795, Dst Port: 53

1. Who is the DHCP Discover sent to? Who is the intended recipient?



It is sent to 255.255.255.255. The intended recipient is who it offered to, which has the destination 192.168.0.10

**5. References**

* Thomas Weeks, Director of Technologies, Virginia Tech