Introduction to Network Traffic Analysis with Wireshark

Directions for Kali – Hardware on Raspberry Pi

Dr. Shannon Beck, Updated 12/30/21

Introduction

Lab Description: In this lab, you will learn some basics about networks using Wireshark. Wireshark is a very useful tool for network forensics–viewing and storing (capturing) networking traffic. Wireshark is software that is called a packet analyzer. It enables you to view pieces of data (called packets) in real-time as they go in and out of a system. You can save these packets as packet capture (pcap or cap) files. In this lab, you will capture live network traffic in real-time and analyze previously captured packet files.

Objectives

- Use Wireshark and gain familiarity with the software tool
- Perform network packet capture through Wireshark
- Inspect packet capture (pcap) files in Wireshark
- Lean more about DNS, MAC addresses and WLANs

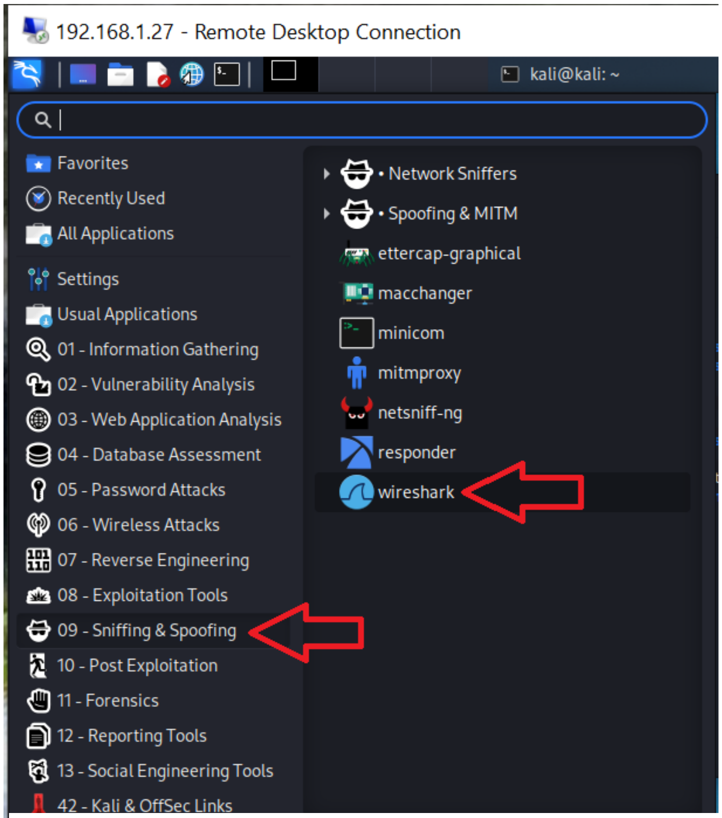


Figure 1: Starting Wireshark through GUI

Lab Environment: Kali Linux

Setting Up the Environment

1) Log on to your Kali Linux virtual machine using the username **kali** and the password **kali**
2) Navigate to “09 ->Sniffing & Spoofing” -> wireshark
3) When prompted to run Wireshark with root permissions, enter the password **kali** and press Enter

Alternative:
Open up a terminal or command-line interface (CLI). Referred to as the QTerminal in the Application menu, or you can click on the black computer screen terminal icon to launch the terminal window. Launch Wireshark.

From the terminal window type **`sudo wireshark`**

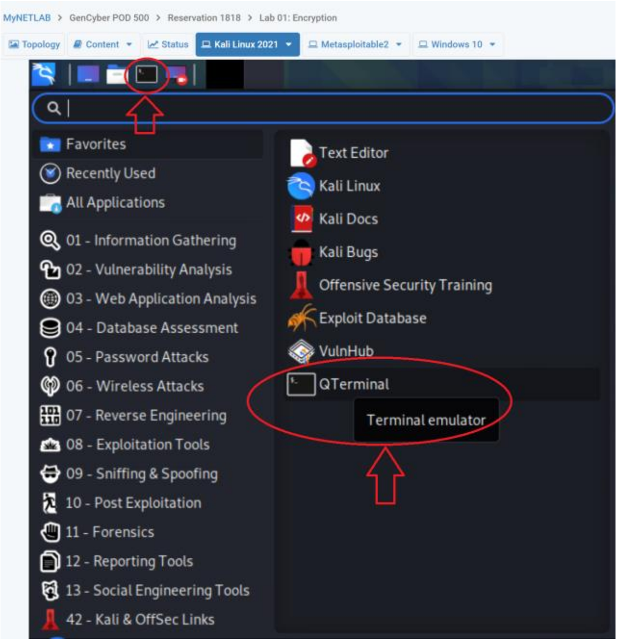
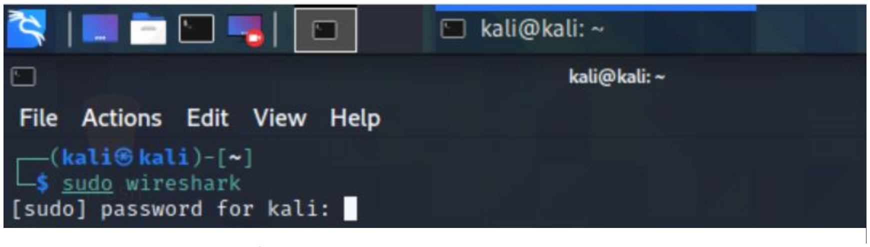


Figure 2: Starting QTerminal



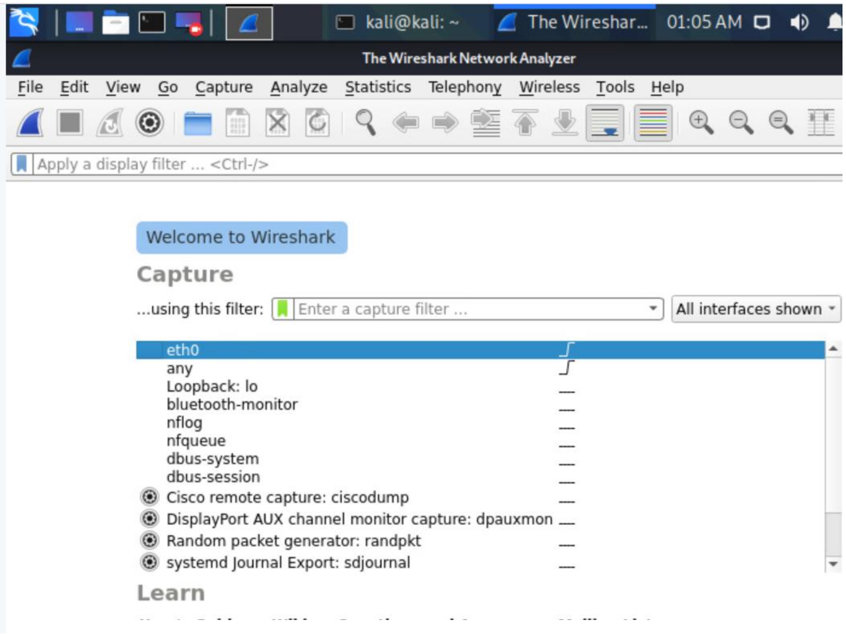
```
┌──(kali㉿kali)-[~]
└─$ sudo wireshark
[sudo] password for kali:
```

Figure 3: Starting Wireshark from CLI

When prompted to run Wireshark with root permissions, enter the password **kali** and press Enter.

If successful, Wireshark should open. Note: It may be necessary to launch this through the terminal window and not the menu system. This ensures that Wireshark opens with the needed administrative permission (using the **sudo** command) so that you can capture traffic.

If Wireshark does not start, open a QTerminal and type the **`sudo wireshark`** and enter the “kali” password.


kali@kali: ~
The Wireshar...
01:05 AM
The Wireshark Network Analyzer
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
Welcome to Wireshark
Capture
Enter a capture filter ...
All interfaces shown
Learn

...using this filter:

eth0
any
Loopback: lo
bluetooth-monitor
nflog
nfqueue
dbus-system
dbus-session
Cisco remote capture: ciscodump
DisplayPort AUX channel monitor capture: dpauxmon
Random packet generator: randpkt
systemd Journal Export: sdjournal

Figure 4: Wireshark interface when first started

Wireshark is a powerful analysis tool that allows you to capture network and device packets and 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 port*, *destination IP*, and other data that we will see in this lab.

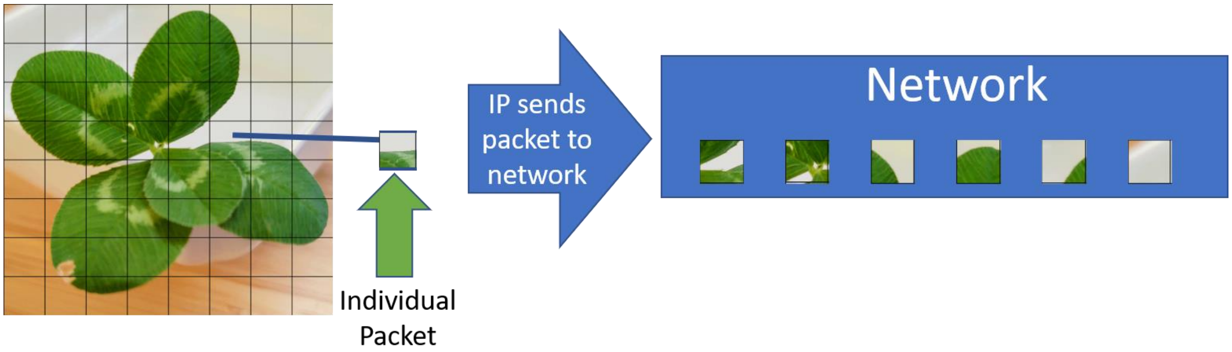
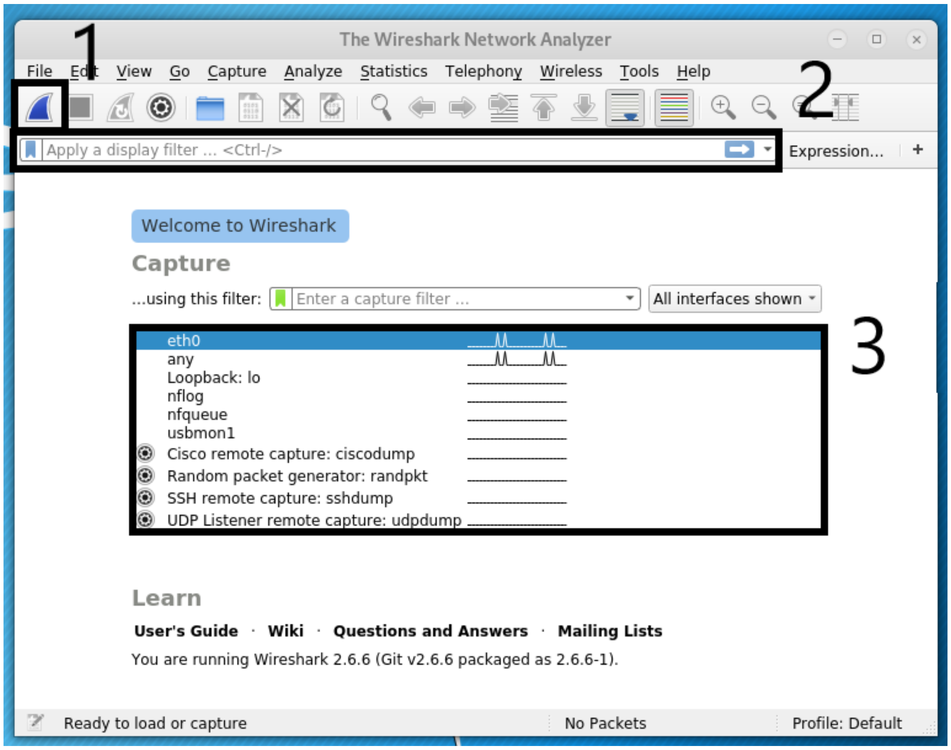


Figure 5: Concept of a packet sent across a network

Let’s explore the Wireshark application interface (not running any commands yet).


1
2
3
The Wireshark Network Analyzer
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
Expression...
Welcome to Wireshark
Capture
...using this filter:
Enter a capture filter ...
All interfaces shown
eth0
any
Loopback: lo
nflog
nfqueue
usbmon1
Cisco remote capture: ciscodump
Random packet generator: randpkt
SSH remote capture: sshdump
UDP Listener remote capture: udpdump
Learn
User's Guide · Wiki · Questions and Answers · Mailing Lists
Ready to load or capture
No Packets
Profile: Default

You are running Wireshark 2.6.6 (Git v2.6.6 packaged as 2.6.6-1).

Figure 6: Wireshark - annotated interface upon startup

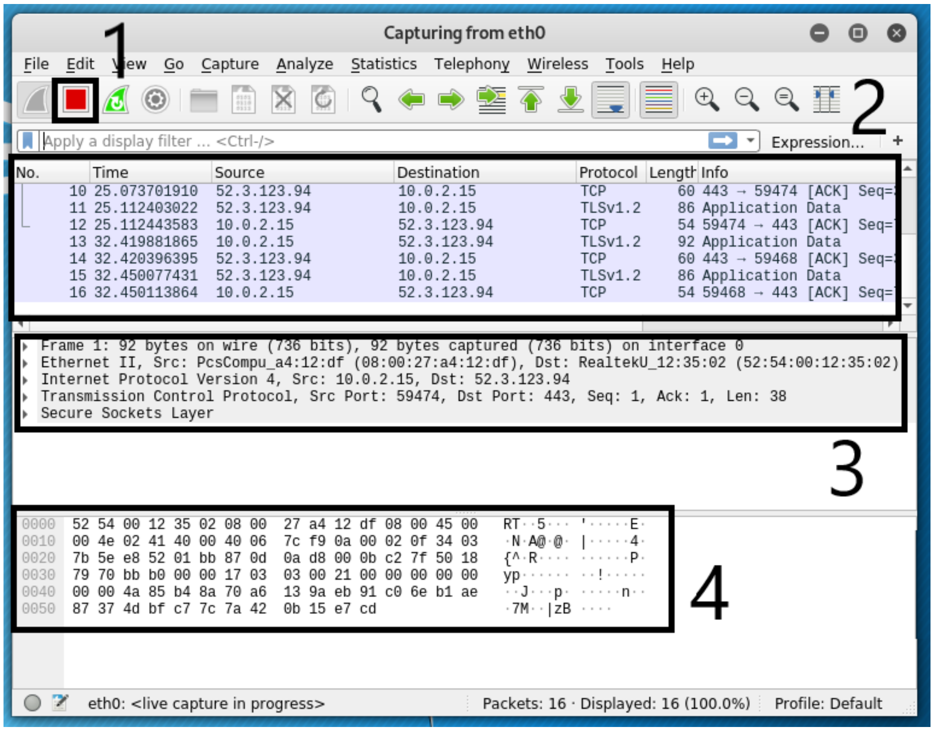
From the above image, we highlight several important Wireshark features through the annotations by the three numbered boxes:

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.
2. **Filter String Field:** This field allows the user to apply filters to the traffic captured. Configure the filters based on your needs:
    - Text in the body of the packet
    - IP address (10.0.0.5)
    - Protocol (i.e. TCP)
    - Port (i.e. 80, 443)

    Filtering is vital when searching through large amounts of data.
3. **Live Packet Devices:** This is a list of the network interfaces from which we can capture traffic. This can be an ethernet port or a wireless (wi-fi) card.

Once Wireshark starts capturing network traffic or opens a file, the screen will change.

Looking at Figure 7, we point out a few more important Wireshark features:



Capturing from eth0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>
Expression…
1
2
3
4

```
No.  Time          Source        Destination   Protocol Length Info
  10 25.073701910  52.3.123.94   10.0.2.15     TCP       60 443 → 59474 [ACK] Seq=
  11 25.112403022  52.3.123.94   10.0.2.15     TLSv1.2   86 Application Data
  12 25.112443583  10.0.2.15     52.3.123.94   TCP       54 59474 → 443 [ACK] Seq=
  13 32.419881865  10.0.2.15     52.3.123.94   TLSv1.2   92 Application Data
  14 32.420396395  52.3.123.94   10.0.2.15     TCP       60 443 → 59468 [ACK] Seq=
  15 32.450077431  52.3.123.94   10.0.2.15     TLSv1.2   86 Application Data
  16 32.450113864  10.0.2.15     52.3.123.94   TCP       54 59468 → 443 [ACK] Seq=
```

```
▸ Frame 1: 92 bytes on wire (736 bits), 92 bytes captured (736 bits) on interface 0
▸ Ethernet II, Src: PcsCompu_a4:12:df (08:00:27:a4:12:df), Dst: RealtekU_12:35:02 (52:54:00:12:35:02)
▸ Internet Protocol Version 4, Src: 10.0.2.15, Dst: 52.3.123.94
▸ Transmission Control Protocol, Src Port: 59474, Dst Port: 443, Seq: 1, Ack: 1, Len: 38
▸ Secure Sockets Layer
```

```
0000  52 54 00 12 35 02 08 00  27 a4 12 df 08 00 45 00   RT··5···  '·····E·
0010  00 4e 02 41 40 00 40 06  7c f9 0a 00 02 0f 34 03   ·N·A@·@·  |·····4·
0020  7b 5e e8 52 01 bb 87 0d  0a d8 00 0b c2 7f 50 18   {^·R····  ······P·
0030  79 70 bb b0 00 00 17 03  03 00 21 00 00 00 00 00   yp······  ··!·····
0040  00 00 4a 85 b4 8a 70 a6  13 9a eb 91 c0 6e b1 ae   ··J···p·  ·····n··
0050  87 37 4d bf c7 7c 7a 42  0b 15 e7 cd               ·7M··|zB  ····
```

eth0: <live capture in progress>   Packets: 16 · Displayed: 16 (100.0%)   Profile: Default

Figure 7: The three main inspection fames from a capture in Wireshark

1. **Stop Capture Button:** This button stops the current capture if you are collecting live network traffic. Once you click this, you can analyze the data and have the option to save it as a .pcap file (a file containing captured packet data) for further analysis or exporting.

2. **Packet List Pane:** This displays the captured data packets. It shows the Source IP, Destination IP, Protocol, and more information about each packet.

3. **Packet Details Pane:** After selecting a packet, this pane displays the packet's 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 stack of encapsulated data structures. From the top, the Layer-1 frame and Ethernet data, wrapped in a Layer-2 IP datagram wrapped in a Layer-3 TCP transport session or connection.

4. **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 the basics using Wireshark to capture data, let's look at a few pcap files, interpret the data in them, and see what we can learn!

TASK 1: ANALYZING SQL SLAMMER WORK PCAP FILE

You will look at .pcap (pronounced “p cap”) files for analysis.

The first two files are from the Wireshark website. These samples are available on the Wireshark Wiki page at https://gitlab.com/wireshark/wireshark/-/wikis/SampleCaptures.

- Under heading of Viruses and worms: slammer.pcap
- Under heading of General / Unsorted: dns.cap

The third file will take a look at some wireless (Wi-Fi) access points.

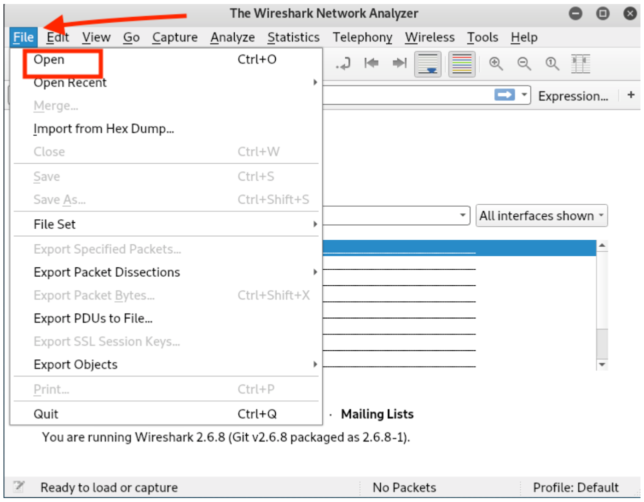
- Fake_Access_Point_Beacons.pcap

The packet capture files are stored in the “Common -> Wireshark” folder on the Desktop.

4) From your running Wireshark program, be sure you have stopped collection (no need to save it

Open the slammer.pcap file. To open it, click on the File menu, then click Open. Navigate to **/home> kali> Desktop> Common> Wireshark**.

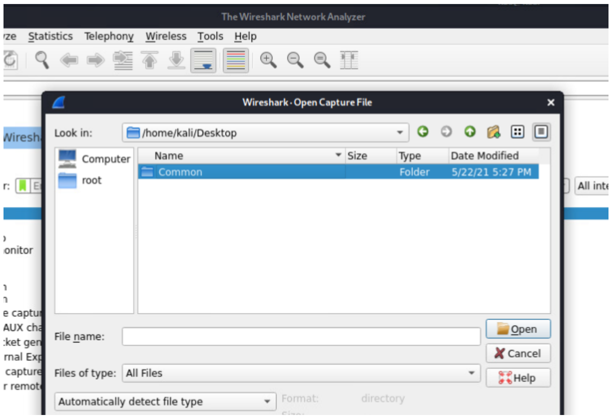
Select **slammer.pcap** and click on Open. This packet capture contains a single packet. The SQL Slammer worm infected hosts through a single packet (!) and spread rapidly.



You are running Wireshark 2.6.8 (Git v2.6.8 packaged as 2.6.8-1).

Ready to load or capture | No Packets | Profile: Default

Figure 8: Opening a p(cap) file for analysis



Look in: /home/kali/Desktop

Common Folder 5/22/21 5:27 PM

File name:

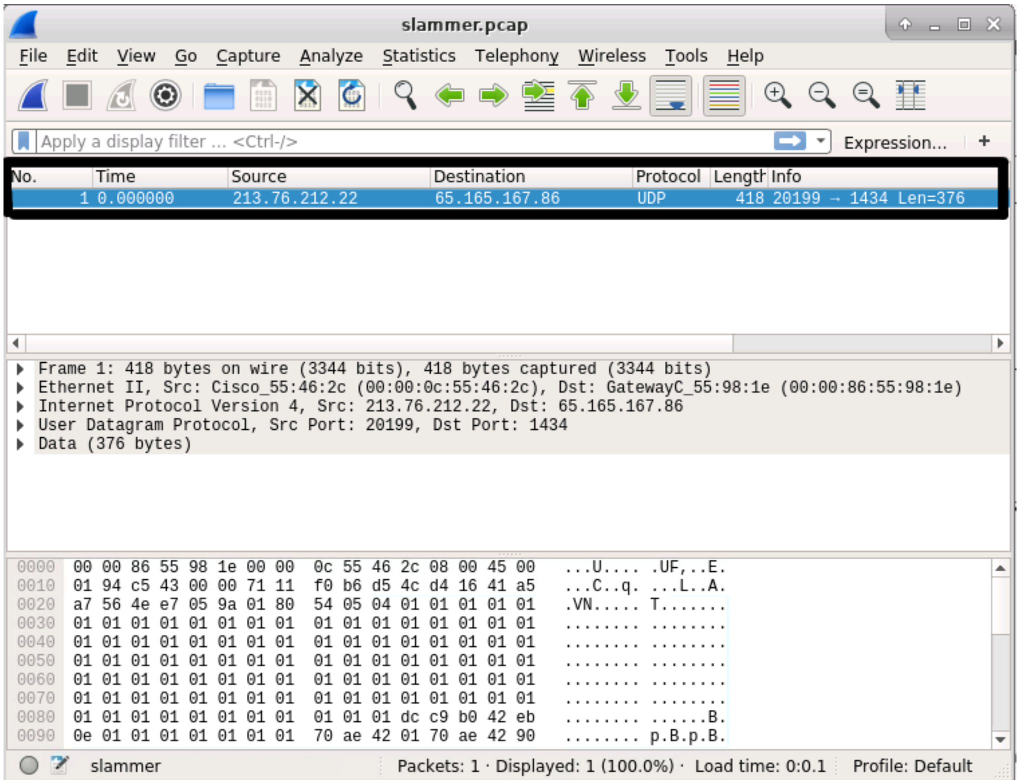
Files of type: All Files

Automatically detect file type

Format: directory

Figure 9: Navigating the Desktop>Common>Wireshark folders

5) Looking at the top packet in the Packet List Pane in the figure below, let’s see what information we can gather


slammer.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>
Expression…
+

```
No.   Time        Source          Destination     Protocol Length Info
    1 0.000000    213.76.212.22   65.165.167.86   UDP         418 20199 → 1434 Len=376
```

```
▸ Frame 1: 418 bytes on wire (3344 bits), 418 bytes captured (3344 bits)
▸ Ethernet II, Src: Cisco_55:46:2c (00:00:0c:55:46:2c), Dst: GatewayC_55:98:1e (00:00:86:55:98:1e)
▸ Internet Protocol Version 4, Src: 213.76.212.22, Dst: 65.165.167.86
▸ User Datagram Protocol, Src Port: 20199, Dst Port: 1434
▸ Data (376 bytes)
```

```
0000  00 00 86 55 98 1e 00 00  0c 55 46 2c 08 00 45 00   ...U.... .UF,..E.
0010  01 94 c5 43 00 00 71 11  f0 b6 d5 4c d4 16 41 a5   ...C..q. ...L..A.
0020  a7 56 4e e7 05 9a 01 80  54 05 04 01 01 01 01 01   .VN..... T.......
0030  01 01 01 01 01 01 01 01  01 01 01 01 01 01 01 01   ........ ........
0040  01 01 01 01 01 01 01 01  01 01 01 01 01 01 01 01   ........ ........
0050  01 01 01 01 01 01 01 01  01 01 01 01 01 01 01 01   ........ ........
0060  01 01 01 01 01 01 01 01  01 01 01 01 01 01 01 01   ........ ........
0070  01 01 01 01 01 01 01 01  01 01 01 01 01 01 01 01   ........ ........
0080  01 01 01 01 01 01 01 01  01 01 01 dc c9 b0 42 eb   ........ ......B.
0090  0e 01 01 01 01 01 01 01  70 ae 42 01 70 ae 42 90   ........ p.B.p.B.
```

```
slammer    Packets: 1 · Displayed: 1 (100.0%) · Load time: 0:0.1    Profile: Default
```

Figure 10: Inspecting the 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. The sender’s address.
Destination	The IP address of the intended recipient of the packet. The receiver’s address.
Protocol	The networking protocol used to send this packet. In Wireshark, if we desire we can filter captured data based on specific protocols. Specifically, this data is Network Layer traffic and

	we can see this packet is using the User Datagram Protocol (UDP) protocol as opposed to TCP traffic.
Length	The data length/size of the packet.
Info	Details about the packet; sometimes helpful.

The Packet Details pane allows you to analyze the packet highlighted in the Packet List Pane.

Use the Packet Details Pane to analyze the details of the **slammer.pcap** file to answer the questions below.

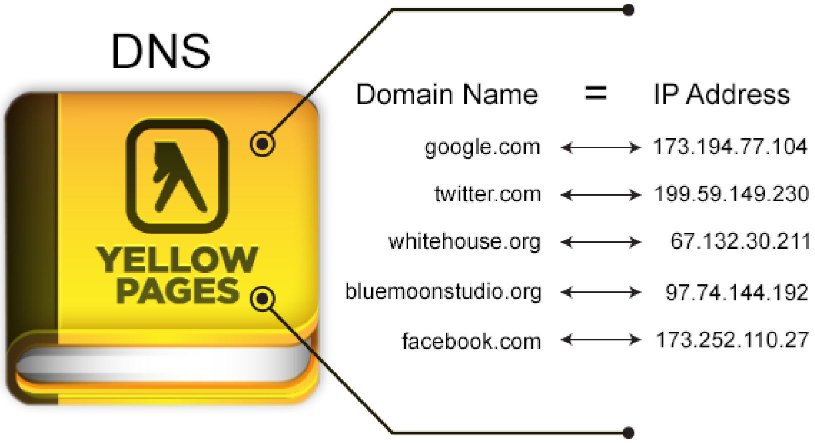
QUESTION 1: What is the *DESTINATION port* of the packet in question? Hint the DESTINATION IP address IS 65.165.167.86. The SQL Server Browser service listens on this port for incoming connections.

QUESTION 2: Name the protocol used to spread the SQL Slammer Worm in this packet.

EXTENSION ACTIVITY: Explore the web to learn more about the slammer worm. What type of applications did it impact and target? What year did it first appear?

TASK 2: ANALYZING DNS PACKETS

A brief overview of DNS


DNS
YELLOW PAGES
Domain Name = IP Address
google.com ⟷ 173.194.77.104
twitter.com ⟷ 199.59.149.230
whitehouse.org ⟷ 67.132.30.211
bluemoonstudio.org ⟷ 97.74.144.192
facebook.com ⟷ 173.252.110.27

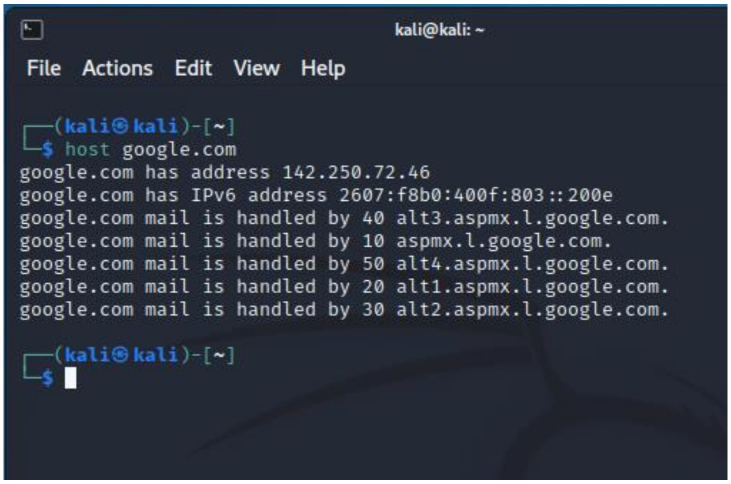
When you enter “google.com” into the web browser, this name has to be translated into a (destination) IP address. This allows the TCP/IP protocol to connect your web browser to a Google server and serve the content that the Google server provides.

This text name translation from

google.com to an IP address is completed through the Domain Name System (DNS). Think of having a phonebook lookup for webpages. The text URL is the "name" and the IP address is like the phone number in a phonebook.

You can manually look up the DNS record to translate google.com into its IP address by typing the DNS lookup command at the CLI or terminal. Open up QTerminal Emulator if you have not already and type:

```
host google.com
```



```
┌──(kali㉿kali)-[~]
└─$ host google.com
google.com has address 142.250.72.46
google.com has IPv6 address 2607:f8b0:400f:803::200e
google.com mail is handled by 40 alt3.aspmx.l.google.com.
google.com mail is handled by 10 aspmx.l.google.com.
google.com mail is handled by 50 alt4.aspmx.l.google.com.
google.com mail is handled by 20 alt1.aspmx.l.google.com.
google.com mail is handled by 30 alt2.aspmx.l.google.com.

┌──(kali㉿kali)-[~]
└─$ 
```

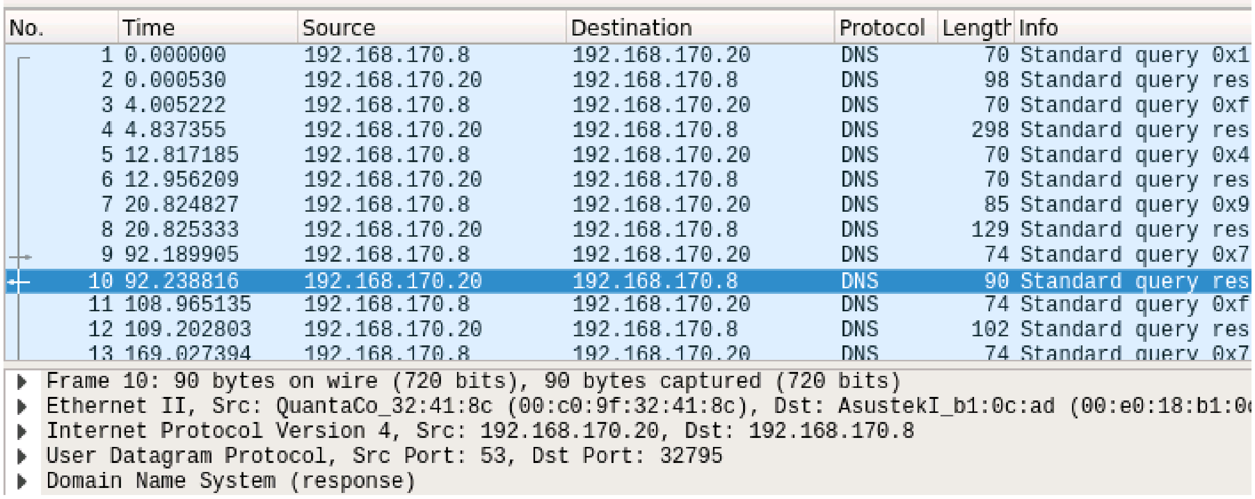
This will resolve (translate) the domain google.com to one or more IPv4 addresses (called A records), IPv6 addresses (called AAAA records), and also provide the known email servers that handle email traffic for that domain (called mail exchange (MX) records). This type of DNS resolution happens every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

To understand how the Domain Name System (DNS) works, we will explore the **dns.cap** packet capture file.

6) Open the **dns.cap** file by selecting Wireshark's **File / Open or Ctrl+O**. It is in the same **/home/kali/Desktop/Common/Wireshark** directory

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

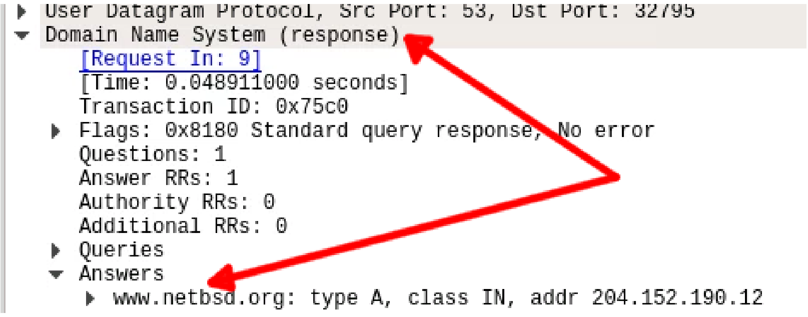
Hint: you can resize the window to make it easier to view the packet data.



```
No.      Time          Source            Destination       Protocol  Length Info
       1 0.000000      192.168.170.8     192.168.170.20    DNS           70 Standard query 0x1
       2 0.000530      192.168.170.20    192.168.170.8     DNS           98 Standard query res
       3 4.005222      192.168.170.8     192.168.170.20    DNS           70 Standard query 0xf
       4 4.837355      192.168.170.20    192.168.170.8     DNS          298 Standard query res
       5 12.817185     192.168.170.8     192.168.170.20    DNS           70 Standard query 0x4
       6 12.956209     192.168.170.20    192.168.170.8     DNS           70 Standard query res
       7 20.824827     192.168.170.8     192.168.170.20    DNS           85 Standard query 0x9
       8 20.825333     192.168.170.20    192.168.170.8     DNS          129 Standard query res
       9 92.189905     192.168.170.8     192.168.170.20    DNS           74 Standard query 0x7
      10 92.238816     192.168.170.20    192.168.170.8     DNS           90 Standard query res
      11 108.965135    192.168.170.8     192.168.170.20    DNS           74 Standard query 0xf
      12 109.202803    192.168.170.20    192.168.170.8     DNS          102 Standard query res
      13 169.027394    192.168.170.8     192.168.170.20    DNS           74 Standard query 0x7
▸ Frame 10: 90 bytes on wire (720 bits), 90 bytes captured (720 bits)
▸ Ethernet II, Src: QuantaCo_32:41:8c (00:c0:9f:32:41:8c), Dst: AsustekI_b1:0c:ad (00:e0:18:b1:0c
▸ Internet Protocol Version 4, Src: 192.168.170.20, Dst: 192.168.170.8
▸ User Datagram Protocol, Src Port: 53, Dst Port: 32795
▸ Domain Name System (response)
```

Figure 11: dns.cap file opened in Wireshark

7) In the Packet Details pane, we can examine much more of the packet's internal details. Let's examine the DNS information for packet no. 10. See the figure below



```
▸ User Datagram Protocol, Src Port: 53, Dst Port: 32795
▾ Domain Name System (response)
    [Request In: 9]
    [Time: 0.048911000 seconds]
    Transaction ID: 0x75c0
  ▸ Flags: 0x8180 Standard query response, No error
    Questions: 1
    Answer RRs: 1
    Authority RRs: 0
    Additional RRs: 0
  ▸ Queries
  ▾ Answers
    ▸ www.netbsd.org: type A, class IN, addr 204.152.190.12
```

Figure 12: Packet Details Pane with expanded entries

Expand the **Domain Name System (response)** by clicking on the arrow to the left of the text.

8) Expand the **Answers** entry and you should see an IP response for the domain www.netbsd.org. This means that a DNS query was made for the domain www.netbsd.org and the IP address of the website was successfully retrieved at 204.152.90.12

QUESTION 3: In Packet No. 5, what URL was queried for its DNS information?

Hint: look under "Domain Name System (response)" and expand the "Queries" tab.

Note: the DNS response is in packet 6 (helpfully noted by Wireshark).

QUESTION 4: In Packet No. 5 (and all of these DNS packets), what protocol is used?

QUESTION 5: What is the IP address of the Domain Name Server (i.e. the source computer that provides the responses to the queries)? This is the source IP address of the packets that have the DNS responses (answers) such as in Packets 6 and 8.

TASK 3: ANALYZING WIRELESS PACKETS

Wireless networks have become very common in providing internet connectivity. With this connectivity, come attacks from physically nearby sources. Home wireless networks and large corporate networks are all vulnerable to these attacks. Hackers employ tools to search for unsecured networks in order to steal internet access or to break into networks. In fact, Kali has several tools that can assist in setting up this type of attack.

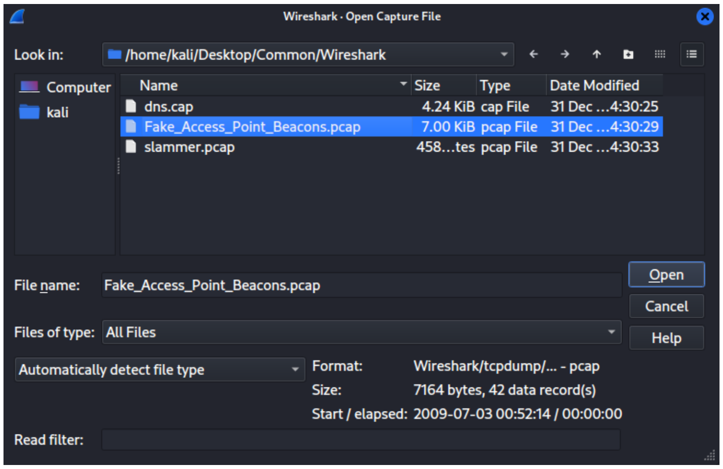
Attackers might accomplish this by going to common hangouts such as restaurants, hotels, or coffee shops that offer free Wi-Fi. Through the use of wireless sniffing tools or even a regular wi-fi interface, unsecured networks are discovered. Attackers can also use hot-spots to set up rouge access points (APs) to fool people into connecting to these access points. The objective of this task is to perform network traffic packet analysis to identify rouge APs.

Find the rouge access point

In this exercise, you will determine if any rouge APs have been deployed in the network.

9) In Wireshark, select **File** then **Open**

10) In the Desktop > Common > Wireshark directory, select the **Fake_Access_Point_Beacon.pcap** file and click **Open**. Wireshark will load this file for you to analyze.



Wireshark · Open Capture File

Look in: /home/kali/Desktop/Common/Wireshark

Name	Size	Type	Date Modified
dns.cap	4.24 KiB	cap File	31 Dec …4:30:25
Fake_Access_Point_Beacons.pcap	7.00 KiB	pcap File	31 Dec …4:30:29
slammer.pcap	458…tes	pcap File	31 Dec …4:30:33

File name: Fake_Access_Point_Beacons.pcap

Files of type: All Files

Automatically detect file type

Format: Wireshark/tcpdump/… - pcap

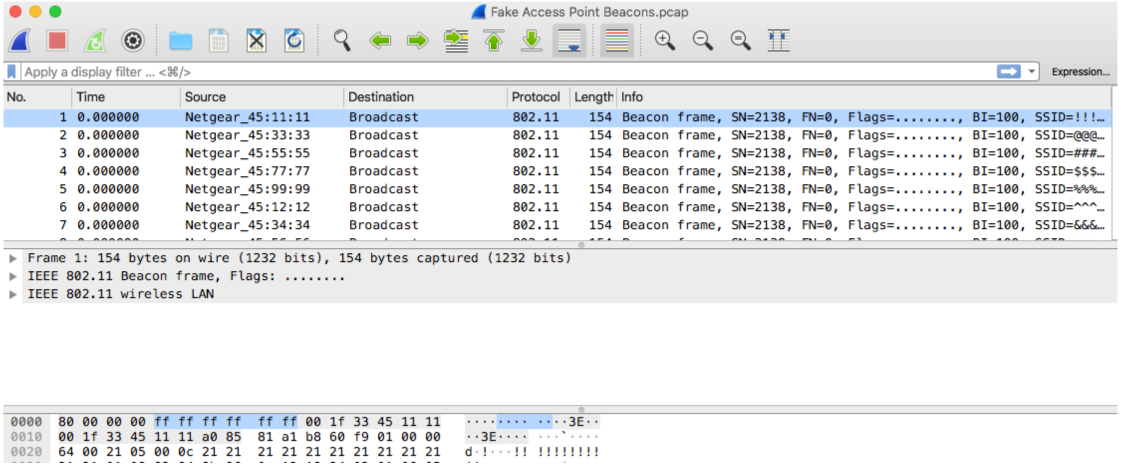
Size: 7164 bytes, 42 data record(s)

Start / elapsed: 2009-07-03 00:52:14 / 00:00:00

Read filter:

Figure 13: Opening Fake_Access_Point_Beacons.pcap

11) Once the file has loaded, you should see the screen below

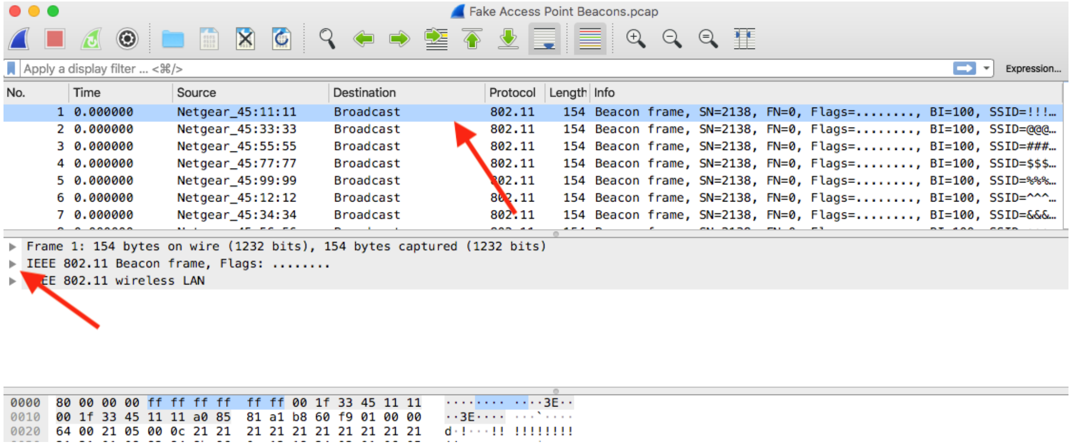

Fake Access Point Beacons.pcap
Apply a display filter ... <⌘/>
Expression...

```
No.  Time      Source             Destination  Protocol  Length  Info
  1  0.000000  Netgear_45:11:11   Broadcast    802.11       154  Beacon frame, SN=2138, FN=0, Flags=........, BI=100, SSID=!!!…
  2  0.000000  Netgear_45:33:33   Broadcast    802.11       154  Beacon frame, SN=2138, FN=0, Flags=........, BI=100, SSID=@@@…
  3  0.000000  Netgear_45:55:55   Broadcast    802.11       154  Beacon frame, SN=2138, FN=0, Flags=........, BI=100, SSID=###…
  4  0.000000  Netgear_45:77:77   Broadcast    802.11       154  Beacon frame, SN=2138, FN=0, Flags=........, BI=100, SSID=$$$…
  5  0.000000  Netgear_45:99:99   Broadcast    802.11       154  Beacon frame, SN=2138, FN=0, Flags=........, BI=100, SSID=%%%…
  6  0.000000  Netgear_45:12:12   Broadcast    802.11       154  Beacon frame, SN=2138, FN=0, Flags=........, BI=100, SSID=^^^…
  7  0.000000  Netgear_45:34:34   Broadcast    802.11       154  Beacon frame, SN=2138, FN=0, Flags=........, BI=100, SSID=&&&…

▶ Frame 1: 154 bytes on wire (1232 bits), 154 bytes captured (1232 bits)
▶ IEEE 802.11 Beacon frame, Flags: ........
▶ IEEE 802.11 wireless LAN

0000  80 00 00 00 ff ff ff ff  ff ff 00 1f 33 45 11 11   ········ ····3E··
0010  00 1f 33 45 11 11 a0 85  81 a1 b8 60 f9 01 00 00   ··3E···· ···`····
0020  64 00 21 05 00 0c 21 21  21 21 21 21 21 21 21 21   d·!···!! !!!!!!!!
```

Figure 14: Wireshark Fake_Access_Point_Beacons.pcap opened in Wireshark

12) Click on one of the lines shown in the top frame.

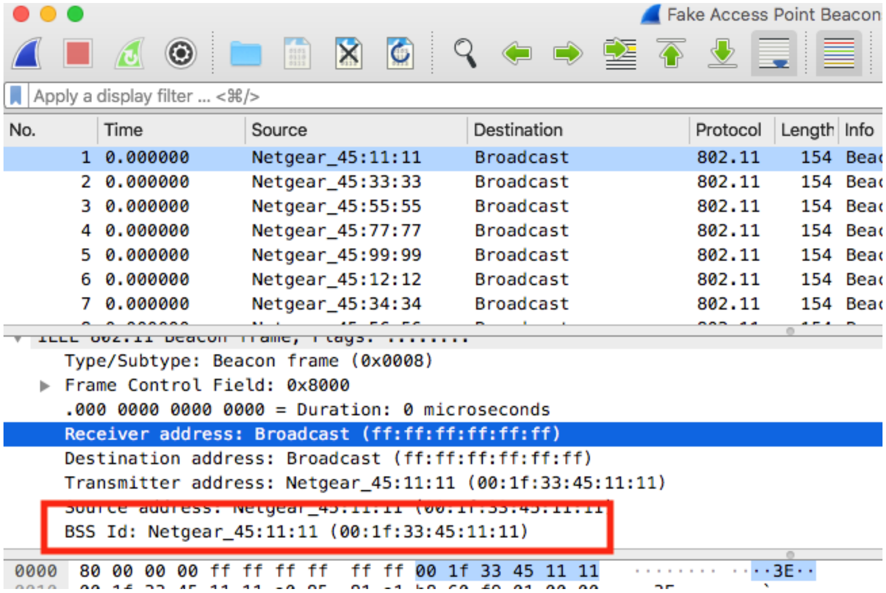
13) Go to the middle frame and click the arrow next to the IEEE 802.11 Beacon frame


Fake Access Point Beacons.pcap
Apply a display filter ... <⌘/>
Expression...

```
No.  Time      Source             Destination  Protocol  Length  Info
  1  0.000000  Netgear_45:11:11   Broadcast    802.11       154  Beacon frame, SN=2138, FN=0, Flags=........, BI=100, SSID=!!!…
  2  0.000000  Netgear_45:33:33   Broadcast    802.11       154  Beacon frame, SN=2138, FN=0, Flags=........, BI=100, SSID=@@@…
  3  0.000000  Netgear_45:55:55   Broadcast    802.11       154  Beacon frame, SN=2138, FN=0, Flags=........, BI=100, SSID=###…
  4  0.000000  Netgear_45:77:77   Broadcast    802.11       154  Beacon frame, SN=2138, FN=0, Flags=........, BI=100, SSID=$$$…
  5  0.000000  Netgear_45:99:99   Broadcast    802.11       154  Beacon frame, SN=2138, FN=0, Flags=........, BI=100, SSID=%%%…
  6  0.000000  Netgear_45:12:12   Broadcast    802.11       154  Beacon frame, SN=2138, FN=0, Flags=........, BI=100, SSID=^^^…
  7  0.000000  Netgear_45:34:34   Broadcast    802.11       154  Beacon frame, SN=2138, FN=0, Flags=........, BI=100, SSID=&&&…

▶ Frame 1: 154 bytes on wire (1232 bits), 154 bytes captured (1232 bits)
▶ IEEE 802.11 Beacon frame, Flags: ........
▶ IEEE 802.11 wireless LAN

0000  80 00 00 00 ff ff ff ff  ff ff 00 1f 33 45 11 11   ········ ····3E··
0010  00 1f 33 45 11 11 a0 85  81 a1 b8 60 f9 01 00 00   ··3E···· ···`····
0020  64 00 21 05 00 0c 21 21  21 21 21 21 21 21 21 21   d·!···!! !!!!!!!!
```

Figure 15: Navigating in Wireshark, inspecting a Beacon Frame

14) Look at the **source address** and **BSSID** (stands for Basic Service Set ID)



```
No.  Time      Source             Destination  Protocol Length Info
  1 0.000000  Netgear_45:11:11   Broadcast    802.11   154  Beac
  2 0.000000  Netgear_45:33:33   Broadcast    802.11   154  Beac
  3 0.000000  Netgear_45:55:55   Broadcast    802.11   154  Beac
  4 0.000000  Netgear_45:77:77   Broadcast    802.11   154  Beac
  5 0.000000  Netgear_45:99:99   Broadcast    802.11   154  Beac
  6 0.000000  Netgear_45:12:12   Broadcast    802.11   154  Beac
  7 0.000000  Netgear_45:34:34   Broadcast    802.11   154  Beac
```

```
▼ IEEE 802.11 Beacon frame, Flags: ........
    Type/Subtype: Beacon frame (0x0008)
  ▶ Frame Control Field: 0x8000
    .000 0000 0000 0000 = Duration: 0 microseconds
    Receiver address: Broadcast (ff:ff:ff:ff:ff:ff)
    Destination address: Broadcast (ff:ff:ff:ff:ff:ff)
    Transmitter address: Netgear_45:11:11 (00:1f:33:45:11:11)
    Source address: Netgear_45:11:11 (00:1f:33:45:11:11)
    BSS Id: Netgear_45:11:11 (00:1f:33:45:11:11)
```

```
0000  80 00 00 00 ff ff ff ff  ff ff 00 1f 33 45 11 11   ........ ..3E..
```

Figure 16: Locating and inspecting the BSS id in Wireshark

15) The first part is the vendor name. The second part in parenthesis is the full MAC (hardware) address. This is way to identify a specific piece of hardware or AP.

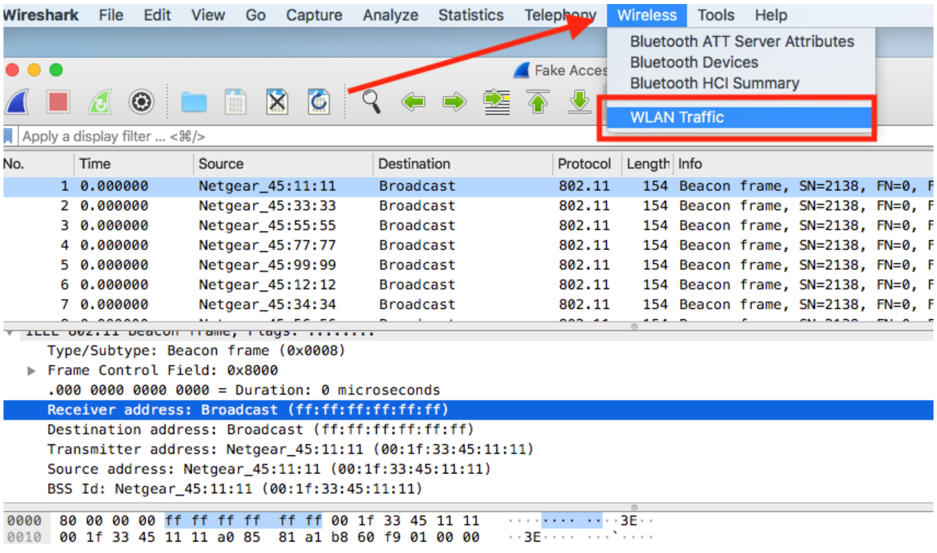
16) Compare the **BSSID**, **vendor name** and **MAC address** against the whitelist.

Device	Vendor	Mac Address	SSID	Channel	BSSID
AP	Netgear	00:1f:33:45:56:56	************	6	Netgear_45:56:56
AP	Cisco-Li	0c:68:03:d6:88:78	Test	48	Cisco-Li_d6:88:78
AP	Netgear	00:1f:33:45:90:90	BBBBBBBBBBBB	6	Netgear_45:90:90

Approved whitelist of good/allowed network devices

BSS Id: Netgear_45:56:56 (00:1f:33:45:56:56)

17) Click on **Wireless > WLAN Traffic**

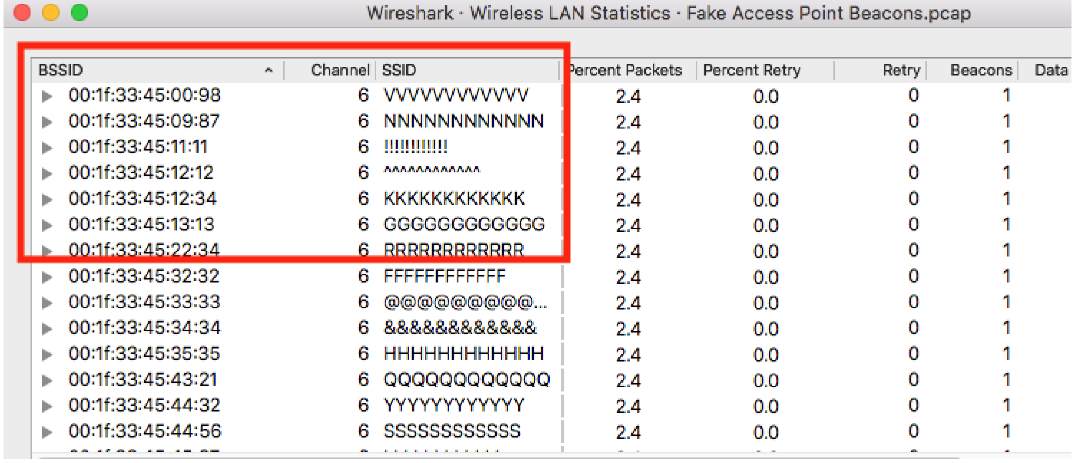


```
No.  Time      Source             Destination  Protocol  Length Info
  1 0.000000  Netgear_45:11:11   Broadcast    802.11     154 Beacon frame, SN=2138, FN=0, F
  2 0.000000  Netgear_45:33:33   Broadcast    802.11     154 Beacon frame, SN=2138, FN=0, F
  3 0.000000  Netgear_45:55:55   Broadcast    802.11     154 Beacon frame, SN=2138, FN=0, F
  4 0.000000  Netgear_45:77:77   Broadcast    802.11     154 Beacon frame, SN=2138, FN=0, F
  5 0.000000  Netgear_45:99:99   Broadcast    802.11     154 Beacon frame, SN=2138, FN=0, F
  6 0.000000  Netgear_45:12:12   Broadcast    802.11     154 Beacon frame, SN=2138, FN=0, F
  7 0.000000  Netgear_45:34:34   Broadcast    802.11     154 Beacon frame, SN=2138, FN=0, F
```

```
    Type/Subtype: Beacon frame (0x0008)
  ▶ Frame Control Field: 0x8000
    .000 0000 0000 0000 = Duration: 0 microseconds
    Receiver address: Broadcast (ff:ff:ff:ff:ff:ff)
    Destination address: Broadcast (ff:ff:ff:ff:ff:ff)
    Transmitter address: Netgear_45:11:11 (00:1f:33:45:11:11)
    Source address: Netgear_45:11:11 (00:1f:33:45:11:11)
    BSS Id: Netgear_45:11:11 (00:1f:33:45:11:11)

0000  80 00 00 00 ff ff ff ff  ff ff 00 1f 33 45 11 11   ···········3E··
0010  00 1f 33 45 11 11 a0 85  81 a1 b8 60 f9 01 00 00   ··3E·······`····
```

Figure 17: Taking a deeper look at the wi-fi traffic

18) Look at the **BSSID, Channel** and **SSIDs** listed


Wireshark · Wireless LAN Statistics · Fake Access Point Beacons.pcap

BSSID	Channel	SSID	Percent Packets	Percent Retry	Retry	Beacons	Data
▶ 00:1f:33:45:00:98	6	VVVVVVVVVVVV	2.4	0.0	0	1	
▶ 00:1f:33:45:09:87	6	NNNNNNNNNNNN	2.4	0.0	0	1	
▶ 00:1f:33:45:11:11	6	!!!!!!!!!!!!	2.4	0.0	0	1	
▶ 00:1f:33:45:12:12	6	^^^^^^^^^^^^	2.4	0.0	0	1	
▶ 00:1f:33:45:12:34	6	KKKKKKKKKKKK	2.4	0.0	0	1	
▶ 00:1f:33:45:13:13	6	GGGGGGGGGGGG	2.4	0.0	0	1	
▶ 00:1f:33:45:22:34	6	RRRRRRRRRRRR	2.4	0.0	0	1	
▶ 00:1f:33:45:32:32	6	FFFFFFFFFFFF	2.4	0.0	0	1	
▶ 00:1f:33:45:33:33	6	@@@@@@@@@...	2.4	0.0	0	1	
▶ 00:1f:33:45:34:34	6	&&&&&&&&&&&&	2.4	0.0	0	1	
▶ 00:1f:33:45:35:35	6	HHHHHHHHHHHH	2.4	0.0	0	1	
▶ 00:1f:33:45:43:21	6	QQQQQQQQQQQQ	2.4	0.0	0	1	
▶ 00:1f:33:45:44:32	6	YYYYYYYYYYYY	2.4	0.0	0	1	
▶ 00:1f:33:45:44:56	6	SSSSSSSSSSSS	2.4	0.0	0	1	

Figure 18: Inspecting the WLan Traffic

19) Check the **BSSID, Channel** and **SSID** being used against the approved whitelist to answer the questions regarding these items
Hint: You can save this information into a file for analysis by clicking on the "Save as…" button in the bottom right.

Answer the following questions:

QUESTION 6: Give the SSID of two APs.

QUESTION 7: What is the SSID being used for the device with the BSS Id: Netgear_45:44:56 (00:1f:33:45:44:56)

QUESTION 8: What is the channel being used by the WAP with the BSSID Netgear_45:44:56?

QUESTION 9: How many of these APs in the approved device whitelist?

QUESTION 10: Are there any rouge APs (yes/no)? This would be any APs on the network that are NOT on the approved whitelist.

QUESTION 11: If rogue wireless access points (WAPs) exist, how many?

20) If you are not continuing to the extension activity, Quit Kali Linux and end your reservation

EXTENSION ACTIVITY: LIVE CAPTURE WITH WIRESHARK

NOTE: Running Wireshark with root-system privileges and is required for operations that monitor and save (capture) live network traffic. When prompted for the root password, enter **kali** to open and run Wireshark.

Your turn!

21) **Start a Live Network Traffic Capture**

Start traffic capture on the Ethernet (eth0) network interface.

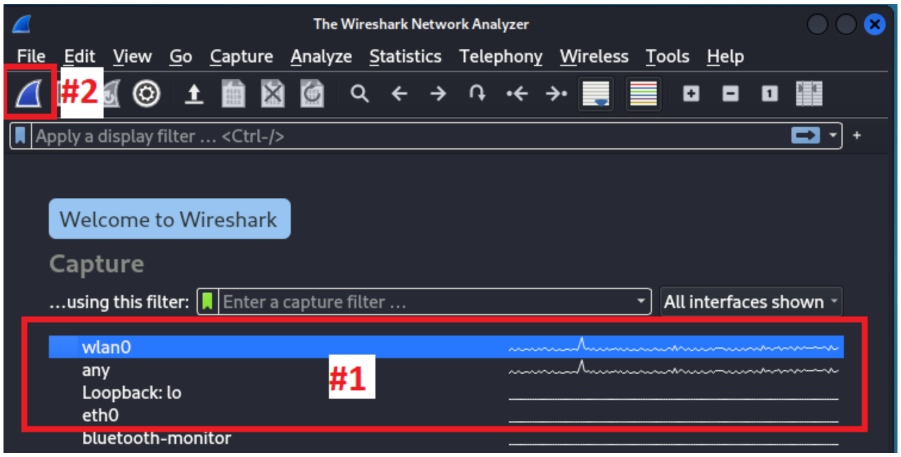


Figure 19: Starting a live capture in Wireshark. Select interface first (#1) and start capture (#2)

Highlight the "**wlan0**" or "**eth0**" interface (Box #1 in Figure 19) depending on which network interface is used to connect to the network on the Raspberry Pi.

Box 3: select "**wlan0**" or "**eth0**" in Box 1. Double clicking on the interface will start the capture.

- OR -

Box 3: select "**wlan0**" or "**eth0**" in Box 1, and then press the shark fin button highlighted by Box #2

After doing so, the live packet capture will record all packet data traversing the host running Wireshark. Observe for a few moments the traffic that is coming in.

> ***WARNING:*** *Capturing live packet traffic takes up a very large amount of memory. Never 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. This can make your system unstable or crash Wireshark (often!). Stop Wireshark by hitting the button next to the #1 as shown in* **Error! Reference source not found.**.

22) To generate network traffic, open a web browser and navigate to a webpage.

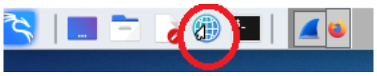
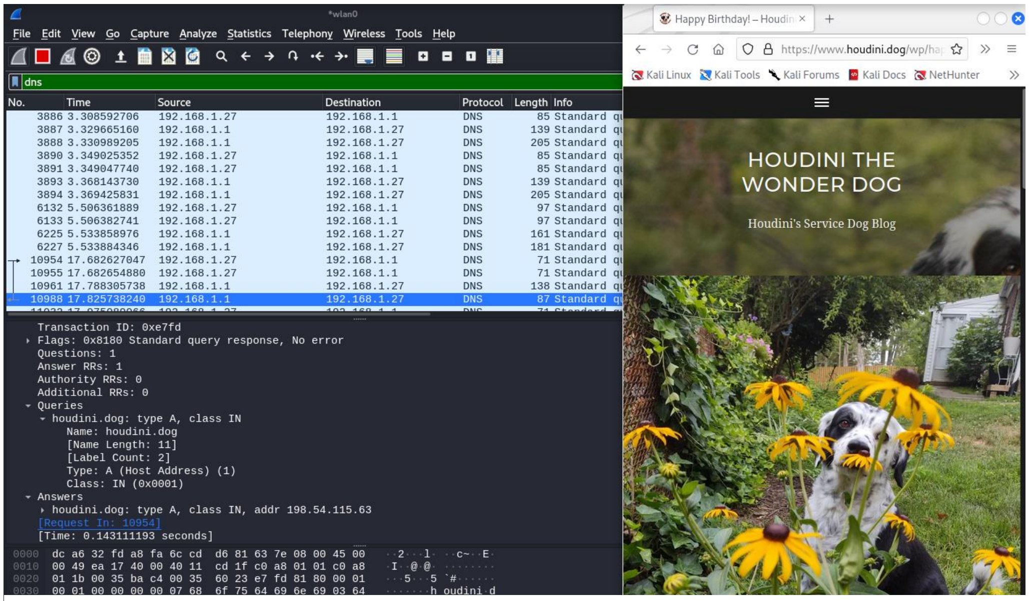


Figure 20: Icon to open up a web browser at top of screen in Kali

23) After capturing some network traffic, stop the capture. To stop the capture, use the red button - OR - under the menu **`Capture -> Stop`**



*wlan0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Happy Birthday! – Houdin ×
https://www.houdini.dog/wp/ha
Kali Linux
Kali Tools
Kali Forums
Kali Docs
NetHunter
HOUDINI THE WONDER DOG
Houdini's Service Dog Blog

```
dns
No.   Time         Source        Destination   Protocol Length Info
 3886 3.308592706  192.168.1.27  192.168.1.1   DNS       85 Standard qu
 3887 3.329665160  192.168.1.1   192.168.1.27  DNS      139 Standard qu
 3888 3.330989205  192.168.1.1   192.168.1.27  DNS      205 Standard qu
 3890 3.349025352  192.168.1.27  192.168.1.1   DNS       85 Standard qu
 3891 3.349047740  192.168.1.27  192.168.1.1   DNS       85 Standard qu
 3893 3.368143730  192.168.1.1   192.168.1.27  DNS      139 Standard qu
 3894 3.369425831  192.168.1.1   192.168.1.27  DNS      205 Standard qu
 6132 5.506361889  192.168.1.27  192.168.1.1   DNS       97 Standard qu
 6133 5.506382741  192.168.1.27  192.168.1.1   DNS       97 Standard qu
 6225 5.533858976  192.168.1.1   192.168.1.27  DNS      161 Standard qu
 6227 5.533884346  192.168.1.1   192.168.1.27  DNS      181 Standard qu
10954 17.682627047 192.168.1.27  192.168.1.1   DNS       71 Standard qu
10955 17.682654880 192.168.1.27  192.168.1.1   DNS       71 Standard qu
10961 17.788305738 192.168.1.1   192.168.1.27  DNS      138 Standard qu
10988 17.825738240 192.168.1.1   192.168.1.27  DNS       87 Standard qu

    Transaction ID: 0xe7fd
  › Flags: 0x8180 Standard query response, No error
    Questions: 1
    Answer RRs: 1
    Authority RRs: 0
    Additional RRs: 0
  ˅ Queries
    ˅ houdini.dog: type A, class IN
        Name: houdini.dog
        [Name Length: 11]
        [Label Count: 2]
        Type: A (Host Address) (1)
        Class: IN (0x0001)
  ˅ Answers
    › houdini.dog: type A, class IN, addr 198.54.115.63
    [Request In: 10954]
    [Time: 0.143111193 seconds]

0000  dc a6 32 fd a8 fa 6c cd  d6 81 63 7e 08 00 45 00   ··2···l· ··c~··E·
0010  00 49 ea 17 40 00 40 11  cd 1f c0 a8 01 01 c0 a8   ·I··@·@· ········
0020  01 1b 00 35 ba c4 00 35  60 23 e7 fd 81 80 00 01   ···5···5 `#······
0030  00 01 00 00 00 00 07 68  6f 75 64 69 6e 69 03 64   ·······h oudini·d
```

Figure 21: Generating web traffic and capturing it in Wireshark

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

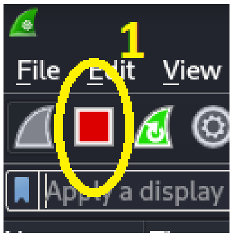


Figure 22: How to stop a live network capture in Wireshark

***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.*

Reflection question: How do the DNS queries and responses relate to the web page that you searched for? Was there more than one DNS entry? Hint: you can filter the DNS packets by entering "DNS" into the filter string field. This field shows as green in Figure 21.

WHAT TO SUBMIT

Submit your answers the questions below.

QUESTION 1: What is the *DESTINATION port* of the packet in question? Hint the DESTINATION IP address IS 65.165.167.86. The SQL Server Browser service listens on this port for incoming connections.

QUESTION 2: Name the protocol used to spread the SQL Slammer Worm in this packet.

QUESTION 3: In Packet No. 5, what URL was queried for its DNS information?

Hint: look under "Domain Name System (response)" and expand the "Queries" tab.

QUESTION 4: In Packet No. 5 (and all of these DNS packets), what protocol is used?

QUESTION 5: What is the IP address of the Domain Name Server (i.e. the computer that handles the phonebook lookups)? This computer is the source of the DNS responses (answers).

QUESTION 6: Give the SSID of two APs.

QUESTION 7: What is the SSID being used for the device with the BSS Id: Netgear_45:44:56 (00:1f:33:45:44:56)

QUESTION 8: What is the channel being used by the WAP with the BSSID Netgear_45:44:56?

QUESTION 9: How many of these APs in the approved device whitelist?

QUESTION 10: Are there any rouge APs (yes/no)? This would be any APs on the network that are NOT on the approved whitelist.

QUESTION 11: If rogue wireless access points (WAPs) exist, how many?

Work adapted from the Virginia Cyber Range Wireshark lab and the 2019 UMGC GenCyber Wireless Lab.