

# **Task 5**

## **Capture and Analyze Network Traffic Using Wireshark.**

### **Introduction:-**

Wireshark is a useful tool for capturing network traffic data. Network pros can make the most of the tool by analyzing captured packets to see what that data means for troubleshooting.

### **Overview**

It's straightforward to capture network traffic with tools such as Wireshark and tcpdump. Many administrators run into trouble when they try to understand what they've captured. This article explains the primary components of captured data and relates this information to the TCP/IP model. The article doesn't cover how to use Wireshark or its features and options. I provide only the basic steps to capture packets enough for you to grab a packet and apply the interpretive information provided throughout.

### **Objective**

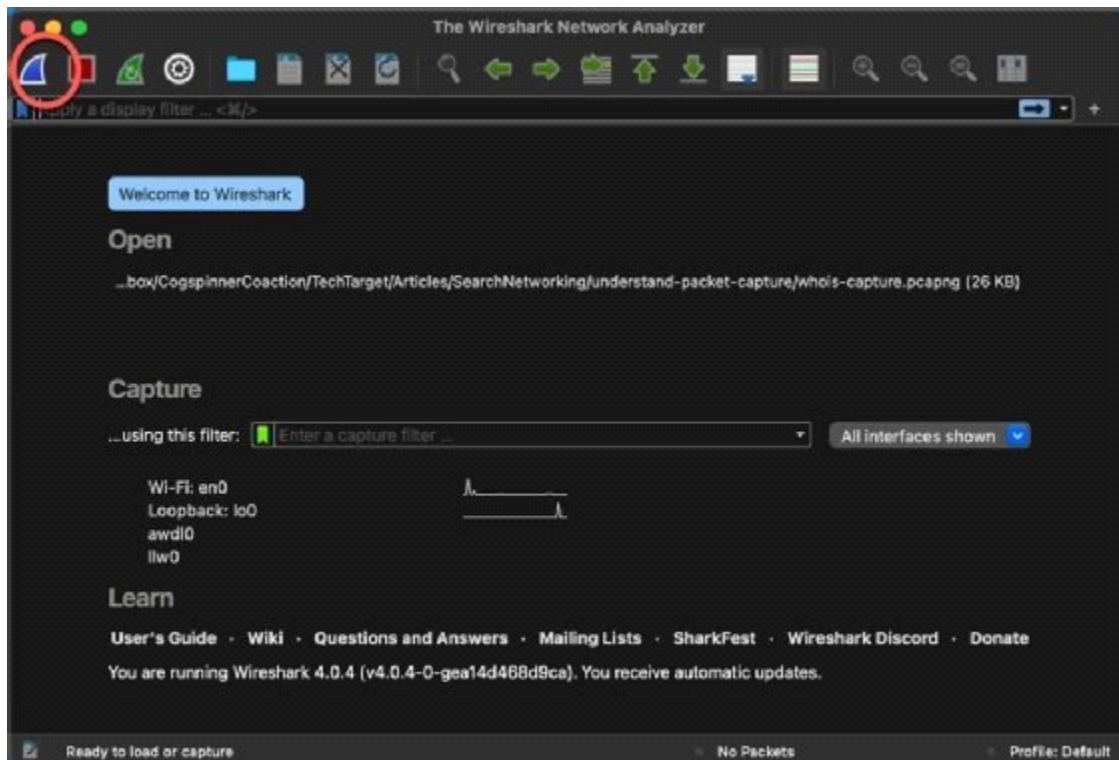
The following steps to initiate a capture in Wireshark:

1. Open Wireshark.
2. Set a capture filter, and select the interface on which to capture.

3.Start the capture.

4.Generate traffic by connecting to a website, pinging a remote device or attempting any other network connection.

5.Stop the capture.

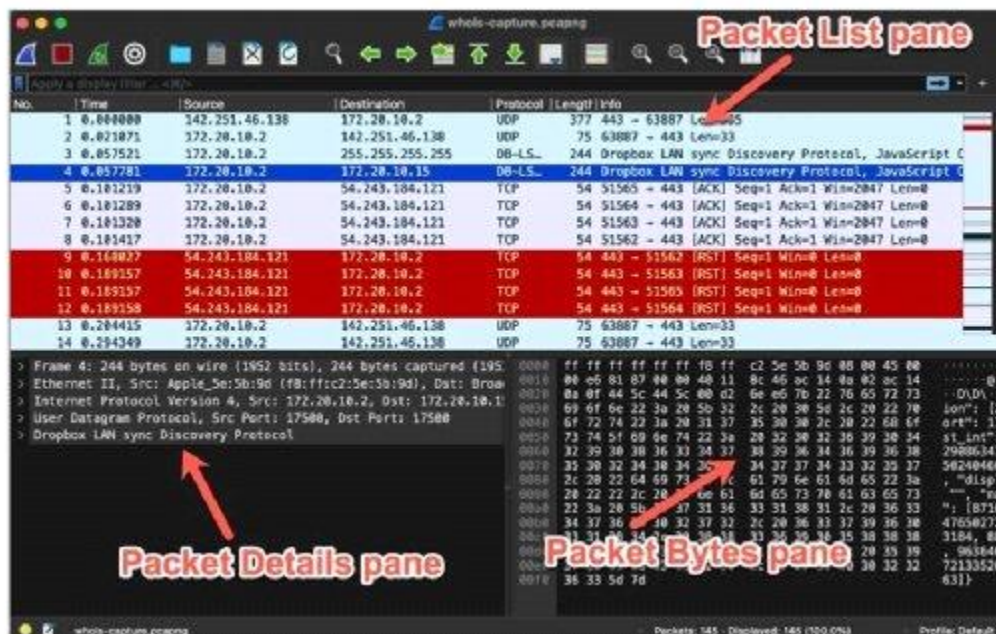


Wireshark captures an immense amount of data quickly if you don't use a filter. While this might be what you want, be sure to set an effective filter if you know the protocols for the service you're troubleshooting. Don't run the capture any longer than you must. Wireshark has various search and filter options, but a targeted capture is much easier to work with.

I use Wireshark in this article because it's common, has a relatively simple GUI and is flexible. But many other powerful protocol analyzers are available, such as tcpdump. You can analyze the content you capture with those tools with the

information below.

Wireshark is powerful and has many options beyond this article's scope, such as network analysis and performance information. Also, Wireshark v3 organizes the output into three vertically stacked window panes. Wireshark v4 uses the same three panes, but the Packet Details pane is in the lower-left corner -- it was the middle pane in v3 -- and the Packet Bytes pane is in the lower-right corner.



## Select a frame in the Packet List pane

In the new Wireshark interface, the top pane summarizes the capture. It displays one or more frames, along with the packet number, time, source, destination, protocol, length and info fields.

Use the protocols, source and destination addresses, and ports columns to help you decide which frame to examine. Click on a frame to select it, and then look at the two lower panes for

details.

The sample capture for these screenshots is a simple Whois query to [www.iana.org](http://www.iana.org). This example generates DNS and other traffic that is handy for the explanations below.

```
damongarn@Damons-MacBook-Pro ~ % whois www.iana.org
% IANA WHOIS server
% for more information on IANA, visit http://www.iana.org
% This query returned 1 object

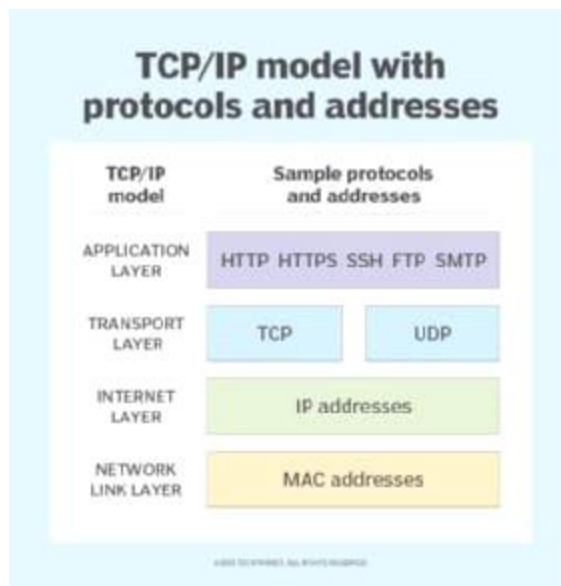
refer:      whois.publicinterestregistry.org

domain:     ORG

organisation: Public Interest Registry (PIR)
address:    11911 Freedom Drive,
address:    10th Floor, Suite 1000
address:    Reston VA 20190
address:    United States of America (the)
```

## Interpret the Packet Details pane

Before I examine the various headers and contents, let's review the TCP/IP model to explain the results in the lower-left Packet Details pane.



Below are the TCP/IP model layers and their associated protocols:

### **Application layer protocols**

HTTP, HTTPS, SMTP, Secure Socket Shell (SSH) and others reside in this layer.

### **Transport layer protocols**

TCP and User Datagram Protocol (UDP) reside in this layer. TCP is connection-oriented (stateful), while UDP is connectionless (stateless). If you capture TCP packets, notice how handshakes that consist of SYN and ACK messages intersperse with the data to ensure all packets arrive at the destination. These units are usually called datagrams.

### **Internet layer protocols**

These protocols govern communication and logical addressing, such as IP addresses. This article focuses on IP, but other protocols exist. These units are usually called packets.

## Network link layer data

This layer displays Ethernet information, including frame types and source and destination MAC addresses. These units are usually called frames.

### Frame content

This frame section provides Ethernet information, such as frame size, time of capture and the physical interface on which the frame was captured.

A screenshot of the Wireshark packet details pane for Frame 71. The pane shows the following information: Section number: 1, Interface id: 0 (en0), Encapsulation type: Ethernet (1), Arrival Time: Mar 24, 2023 12:37:11.964168000 MDT, [Time shift for this packet: 0.000000000 seconds], Epoch Time: 1679683031.964168000 seconds, [Time delta from previous captured frame: 0.073110000 seconds], [Time delta from previous displayed frame: 0.073110000 seconds], [Time since reference or first frame: 1.896159000 seconds], Frame Number: 71, Frame Length: 111 bytes (888 bits), Capture Length: 111 bytes (888 bits), [Frame is marked: False], [Frame is ignored: False], [Protocols in frame: eth:ethertype:ip:udp:dns], [Coloring Rule Name: UDP], [Coloring Rule String: udp].

```
▼ Frame 71: 111 bytes on wire (888 bits), 111 bytes captured (888 bits) on interface en0, id 0
  Section number: 1
    > Interface id: 0 (en0)
      Encapsulation type: Ethernet (1)
      Arrival Time: Mar 24, 2023 12:37:11.964168000 MDT
      [Time shift for this packet: 0.000000000 seconds]
      Epoch Time: 1679683031.964168000 seconds
      [Time delta from previous captured frame: 0.073110000 seconds]
      [Time delta from previous displayed frame: 0.073110000 seconds]
      [Time since reference or first frame: 1.896159000 seconds]
      Frame Number: 71
      Frame Length: 111 bytes (888 bits)
      Capture Length: 111 bytes (888 bits)
      [Frame is marked: False]
      [Frame is ignored: False]
      [Protocols in frame: eth:ethertype:ip:udp:dns]
      [Coloring Rule Name: UDP]
      [Coloring Rule String: udp]
```

Administrators can use this information to examine frame size, for example. Certain devices might have issues accepting frames that exceed the standard size. Troubleshooters can also verify the interface on which the data was captured to ensure information flows through the proper connection.

### Ethernet content

Next is Ethernet II content, including source and destination MAC addresses. Depending on the frame's direction of travel, the local MAC address is either the source or destination

address, and the next network device's MAC address is the other.

```
▼ Ethernet II, Src: e2:2b:96:bc:e0:64 (e2:2b:96:bc:e0:64), Dst: Apple_5e:5b:9d (f8:ff:c2:5e:5b:9d)
  > Destination: Apple_5e:5b:9d (f8:ff:c2:5e:5b:9d)
  > Source: e2:2b:96:bc:e0:64 (e2:2b:96:bc:e0:64)
  Type: IPv4 (0x0800)
```

Confirm the MAC addresses are correct to combat security issues, such as Address Resolution Protocol poisoning or spoofing. Troubleshooters might also confirm which local interface is in use with the MAC address.

## IP content

Next is the IP section, with source and destination IP addresses and port numbers. For most networks, the address structure is IPv4. Time-to-live information exists here, as does fragmentation instructions. Finally, a field defines whether the packet uses TCP or UDP at the transport layer.

```
▼ Internet Protocol Version 4, Src: 172.20.10.1, Dst: 172.20.10.2
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
  Total Length: 97
  Identification: 0x90a1 (37025)
  > 000. .... = Flags: 0x0
  ...0 0000 0000 0000 = Fragment Offset: 0
  Time to Live: 64
  Protocol: UDP (17)
  Header Checksum: 0x7dbf [validation disabled]
  [Header checksum status: Unverified]
  Source Address: 172.20.10.1
  Destination Address: 172.20.10.2
```

Network technicians can verify the IP addresses are valid and expected. Remember, an address beginning with 169.254.x.x is not valid on the network and indicates a possible Dynamic Host

Configuration Protocol problem. Techs can also confirm the source and destination IP addresses are as expected to reduce issues of DNS poisoning or incorrect name resolution settings.

## Transport content

Next is a section containing transport layer information. You should see either TCP or UDP here, depending on the type of datagram captured. Remember, TCP uses the three-way handshake to enumerate the data exchange, which ensures the source device resends any lost data. This section displays the source and destination ports, too. If the packet originates from the client computer, the destination port is the service port number. For example, if the destination system is a web server, the destination port number is 80 or 443 (HTTP or HTTPS). The client's port number is a randomly generated number between 1,024 and 65,535 -- this range varies by OS. Both port numbers appear in this header.

```

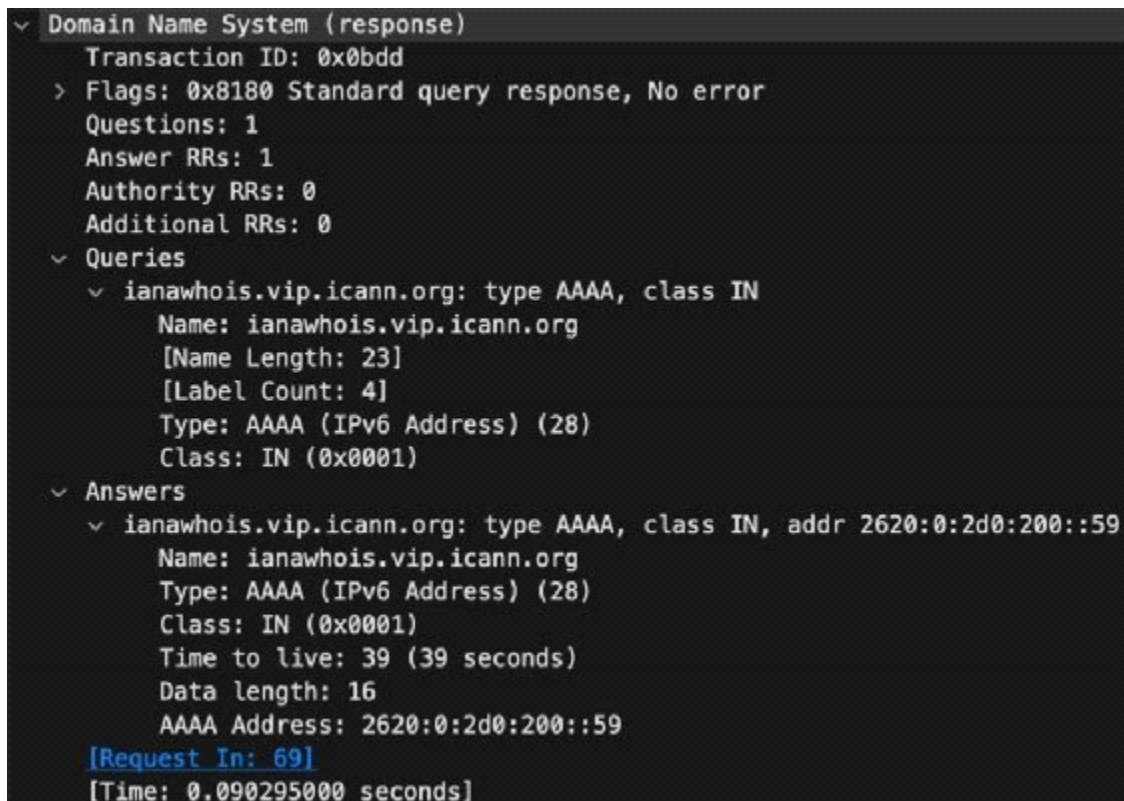
  User Datagram Protocol, Src Port: 53, Dst Port: 56669
    Source Port: 53
    Destination Port: 56669
    Length: 77
    Checksum: 0x0923 [unverified]
    [Checksum Status: Unverified]
    [Stream index: 6]
  > [Timestamps]
    UDP payload (69 bytes)
```

Confirm the client and server use the correct service port number. If you captured packets on the server, this is the destination port number on inbound packets. This is especially important with any nonstandard port numbers for custom applications or odd firewall rules in place.



## Application content

The application layer information is at the bottom of the Packet Details pane but at the top of the TCP/IP model. This information varies by service and protocol. For example, when using HTTP, the pane includes instructions such as GET or the contents of the requested webpage. For capture targeting, you see information with SMTP, Post Office Protocol 3 or Internet Message Access Protocol. The same goes for services such as SSH, network file sharing, DNS, etc.



```

Domain Name System (response)
  Transaction ID: 0x0bdd
  > Flags: 0x8180 Standard query response, No error
  Questions: 1
  Answer RRs: 1
  Authority RRs: 0
  Additional RRs: 0
  Queries
    ianawhois.vip.icann.org: type AAAA, class IN
      Name: ianawhois.vip.icann.org
      [Name Length: 23]
      [Label Count: 4]
      Type: AAAA (IPv6 Address) (28)
      Class: IN (0x0001)
  Answers
    ianawhois.vip.icann.org: type AAAA, class IN, addr 2620:0:2d0:200::59
      Name: ianawhois.vip.icann.org
      Type: AAAA (IPv6 Address) (28)
      Class: IN (0x0001)
      Time to live: 39 (39 seconds)
      Data length: 16
      AAAA Address: 2620:0:2d0:200::59
  [Request In: 69]
  [Time: 0.090295000 seconds]
```

Applications are preconfigured for specific ports, so there probably isn't much room for misconfiguration here. You could use this information to ensure the destination services are running on the server. Wireshark can display data exactly as you'd expect to find it in the TCP/IP model.



