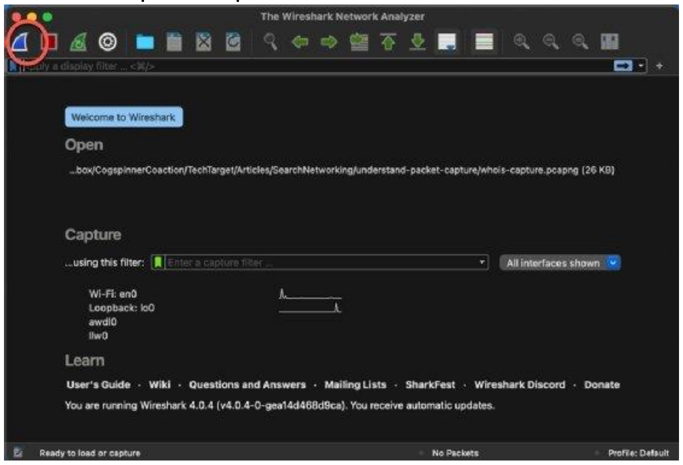
# Examine a captured packet using Wireshark

This article explains the primary components of captured data and relates this information to the TCP/IP model. The article's purpose is not covering 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.
Take 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.

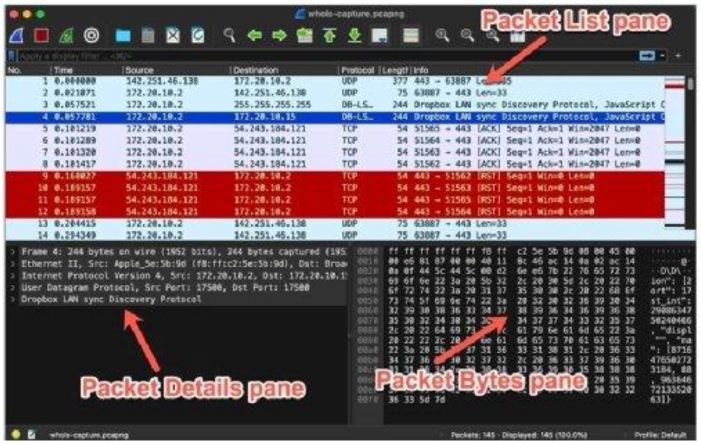

The Wireshark Network Analyzer
Apply a display filter ... <⌘/>
Welcome to Wireshark
Open
Capture
...using this filter:
Enter a capture filter ...
All interfaces shown
Wi-Fi: en0
Loopback: lo0
awdl0
llw0
Learn
User's Guide · Wiki · Questions and Answers · Mailing Lists · SharkFest · Wireshark Discord · Donate
Ready to load or capture
No Packets
Profile: Default

...box/CogspinnerCoaction/TechTarget/Articles/SearchNetworking/understand-packet-capture/whois-capture.pcapng (26 KB)

You are running Wireshark 4.0.4 (v4.0.4-0-gea14d468d9ca). You receive automatic updates.

Start a capture with the shark fin button in the upper left of the Wireshark tool.

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. And 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 is 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 using the information below.
Wireshark is powerful and has many options beyond this article's scope, including network analysis and performance information. Also, note that 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.



```
No.   Time      Source          Destination      Protocol Lengtl Info
    1 0.000000  142.251.46.138  172.20.10.2      UDP      377 443 → 63887 Len=335
    2 0.021071  172.20.10.2     142.251.46.138   UDP       75 63887 → 443 Len=33
    3 0.057521  172.20.10.2     255.255.255.255  DB-LS…   244 Dropbox LAN sync Discovery Protocol, JavaScript C
    4 0.057781  172.20.10.2     172.20.10.15     DB-LS…   244 Dropbox LAN sync Discovery Protocol, JavaScript C
    5 0.101219  172.20.10.2     54.243.184.121   TCP       54 51565 → 443 [ACK] Seq=1 Ack=1 Win=2047 Len=0
    6 0.101289  172.20.10.2     54.243.184.121   TCP       54 51564 → 443 [ACK] Seq=1 Ack=1 Win=2047 Len=0
    7 0.101320  172.20.10.2     54.243.184.121   TCP       54 51563 → 443 [ACK] Seq=1 Ack=1 Win=2047 Len=0
    8 0.101417  172.20.10.2     54.243.184.121   TCP       54 51562 → 443 [ACK] Seq=1 Ack=1 Win=2047 Len=0
    9 0.160027  54.243.184.121  172.20.10.2      TCP       54 443 → 51562 [RST] Seq=1 Win=0 Len=0
   10 0.189157  54.243.184.121  172.20.10.2      TCP       54 443 → 51563 [RST] Seq=1 Win=0 Len=0
   11 0.189157  54.243.184.121  172.20.10.2      TCP       54 443 → 51565 [RST] Seq=1 Win=0 Len=0
   12 0.189158  54.243.184.121  172.20.10.2      TCP       54 443 → 51564 [RST] Seq=1 Win=0 Len=0
   13 0.204415  172.20.10.2     142.251.46.138   UDP       75 63887 → 443 Len=33
   14 0.294349  172.20.10.2     142.251.46.138   UDP       75 63887 → 443 Len=33
```

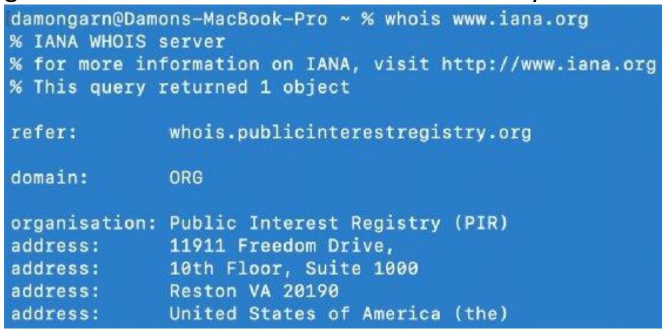
```
> Frame 4: 244 bytes on wire (1952 bits), 244 bytes captured (195
> Ethernet II, Src: Apple_5e:5b:9d (f8:ff:c2:5e:5b:9d), Dst: Broa
> Internet Protocol Version 4, Src: 172.20.10.2, Dst: 172.20.10.1
> User Datagram Protocol, Src Port: 17500, Dst Port: 17500
> Dropbox LAN sync Discovery Protocol
```

```
0000  ff ff ff ff ff ff f8 ff  c2 5e 5b 9d 08 00 45 00   ········
0010  00 e6 81 87 00 00 40 11  8c 46 ac 14 0a 02 ac 14   ······@·
0020  0a 0f 44 5c 44 5c 00 d2  6e e6 7b 22 76 65 72 73   ··D\D\··
0030  69 6f 6e 22 3a 20 5b 32  2c 20 30 5d 2c 20 22 70   ion": [2
0040  6f 72 74 22 3a 20 31 37  35 30 30 2c 20 22 68 6f   ort": 17
0050  73 74 5f 69 6e 74 22 3a  20 32 30 32 36 39 30 34   st_int":
0060  32 39 30 38 36 33 34 37  38 39 36 34 36 39 36 38   29086347
0070  35 30 32 34 30 34 [illegible]  34 37 37 34 33 32 35 37   50240466
0080  2c 20 22 64 69 73 [illegible]  61 79 6e 61 6d 65 22 3a   , "displ
0090  20 22 22 2c 20 [illegible] 6e 61  6d 65 73 70 61 63 65 73    "", "na
00a0  22 3a 20 5b [illegible] 37 31 36  33 31 38 31 2c 20 36 33   ": [8716
00b0  34 37 36 [illegible] 30 32 37 32  2c 20 36 33 37 39 36 30   47650272
[illegible]                                               3184, 88
[illegible]                                      20 35 39   , 963646
[illegible]                                   30 32 32   72133520
00f0  36 33 5d 7d                                         63]}
```

Three panes in the main Wireshark window

**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. 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)
```

A Whois query for www.iana.org

## Interpret the Packet Details pane

Before examining the various headers and contents, I'll briefly review the TCP/IP model to help explain the results displayed in the lower-left Packet Details pane. The TCP/IP model with sample protocols and addresses

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 (or stateful), while UDP is connectionless (or 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*.

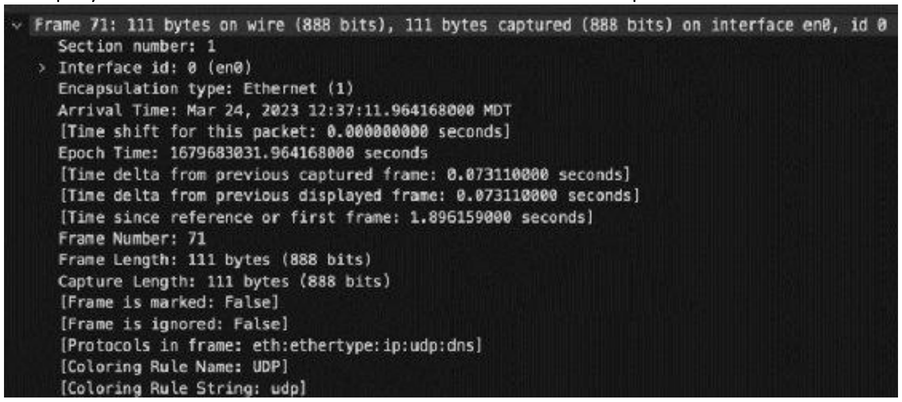
Wireshark displays this output from the bottom of the TCP/IP model upward. Frame information -- the bottom of the TCP/IP model -- resides at the top of the pane in the lower left of the Wireshark screen.

## Understand the headers

How is this information helpful? Relating the headers in the captured frames to the TCP/IP model helps troubleshooters visualize the layers at which problems might occur, helping to identify possible culprits. The following sections address the various layers in more detail.

### 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.



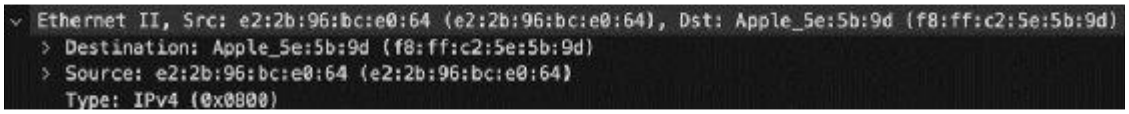
```
v Frame 71: 111 bytes on wire (888 bits), 111 bytes captured (888 bits) on interface en0, id 0
    Section number: 1
  > Interface id: 0 (en0)
    Encapsulation type: Ethernet (1)
    Arrival Time: Mar 24, 2023 12:37:11.964168000 MDT
    [Time shift for this packet: 0.000000000 seconds]
    Epoch 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]
```

Frame information, such as frame size, time of capture and physical interface

**Use this information:** 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 which interface the data was captured on, ensuring 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.



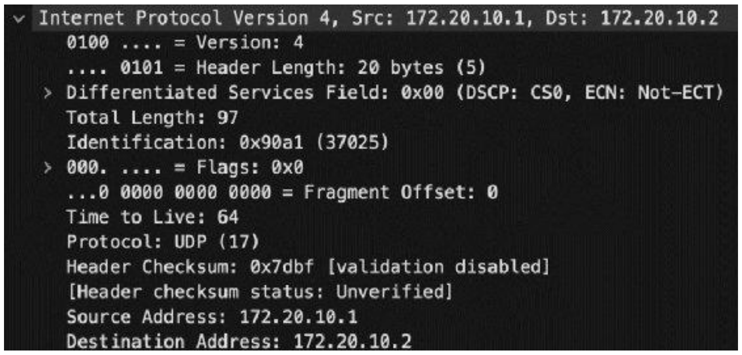
```
v Ethernet II, Src: e2:2b:96:bc:e0:64 (e2:2b:96:bc:e0:64), Dst: Apple_5e:5b:9d (f8:ff:c2:5e:5b:9d)
  > Destination: Apple_5e:5b:9d (f8:ff:c2:5e:5b:9d)
  > Source: e2:2b:96:bc:e0:64 (e2:2b:96:bc:e0:64)
    Type: IPv4 (0x0800)
```

Ethernet information, including MAC addresses

**Use this information:** 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.



```
v 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
```

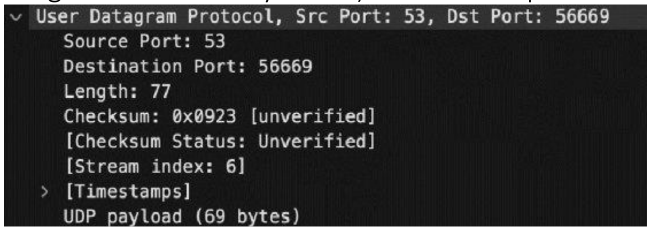
IP information, including IP addresses

**Use this information:** 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, ensuring that 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.



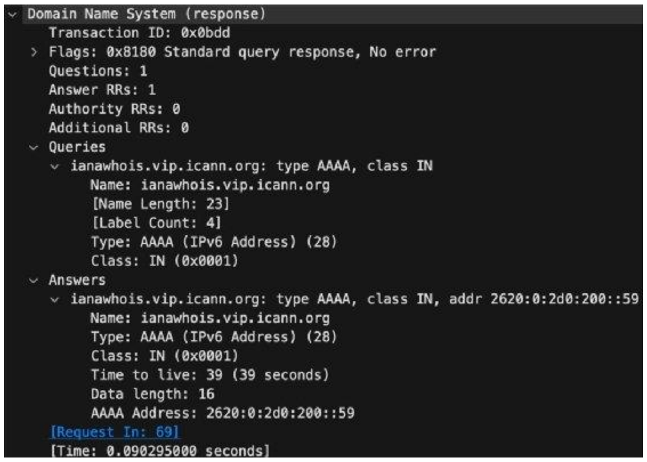
```
v User Datagram Protocol, Src Port: 53, Dst Port: 56669
    Source Port: 53
    Destination Port: 56669
    Length: 77
    Checksum: 0x0923 [unverified]
    [Checksum Status: Unverified]
    [Stream index: 6]
  > [Timestamps]
    UDP payload (69 bytes)
```

UDP content, including destination port and payload

**Use this information:** 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.



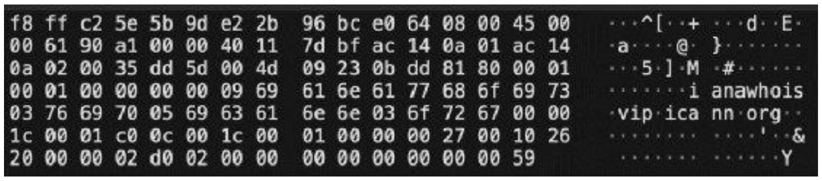
```
v Domain Name System (response)
    Transaction ID: 0x0bdd
  > Flags: 0x8180 Standard query response, No error
    Questions: 1
    Answer RRs: 1
    Authority RRs: 0
    Additional RRs: 0
  v Queries
    v ianawhois.vip.icann.org: type AAAA, class IN
        Name: ianawhois.vip.icann.org
        [Name Length: 23]
        [Label Count: 4]
        Type: AAAA (IPv6 Address) (28)
        Class: IN (0x0001)
  v Answers
    v 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]
```

DNS name resolution content, such as type and class

**Use this information:** 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 does a great job of displaying data exactly as you'd expect to find it in the TCP/IP model.
**Challenge:** Can you relate the content of the captured packet to the seven layers of the OSI model? Each of the layers is represented in the captured information.
The header information in the capture helps confirm that addresses, ports and other settings meet expectations. Such captures tell what is happening on the network, not what should be happening.

## Interpret payload information

The Packet Bytes pane in the lower-right corner of Wireshark displays the payload. This content can be the end-user data security professionals worry about. Unencrypted protocols, such as HTTP, Telnet, SMTP and others, don't protect the confidentiality of their payload, so the data is shown in this window. I frequently demonstrated this in my tech courses using Telnet -- the password was displayed in this pane. Can you find the Whois query in the content example below?



```
f8 ff c2 5e 5b 9d e2 2b  96 bc e0 64 08 00 45 00   ···^[··+ ···d··E·
00 61 90 a1 00 00 40 11  7d bf ac 14 0a 01 ac 14   ·a····@· }·······
0a 02 00 35 dd 5d 00 4d  09 23 0b dd 81 80 00 01   ···5·]·M ·#······
00 01 00 00 00 00 09 69  61 6e 61 77 68 6f 69 73   ·······i anawhois
03 76 69 70 05 69 63 61  6e 6e 03 6f 72 67 00 00   ·vip·ica nn·org··
1c 00 01 c0 0c 00 1c 00  01 00 00 00 27 00 10 26   ········ ····'··&
20 00 00 02 d0 02 00 00  00 00 00 00 00 00 59      ········ ······Y
```

Payload content in cleartext

The payload information demonstrates the importance of HTTPS, SSH and other protocols that encrypt data -- or the use of IPsec, which can encrypt protocols that don't offer built-in encryption themselves.