

Wireshark para novatos - Español



Wireshark para novatos Por Anmol K Sachan

anmol221999@gmail.com

Linkedin: <https://linkedin.com/in/anmolksachan/> Ig:

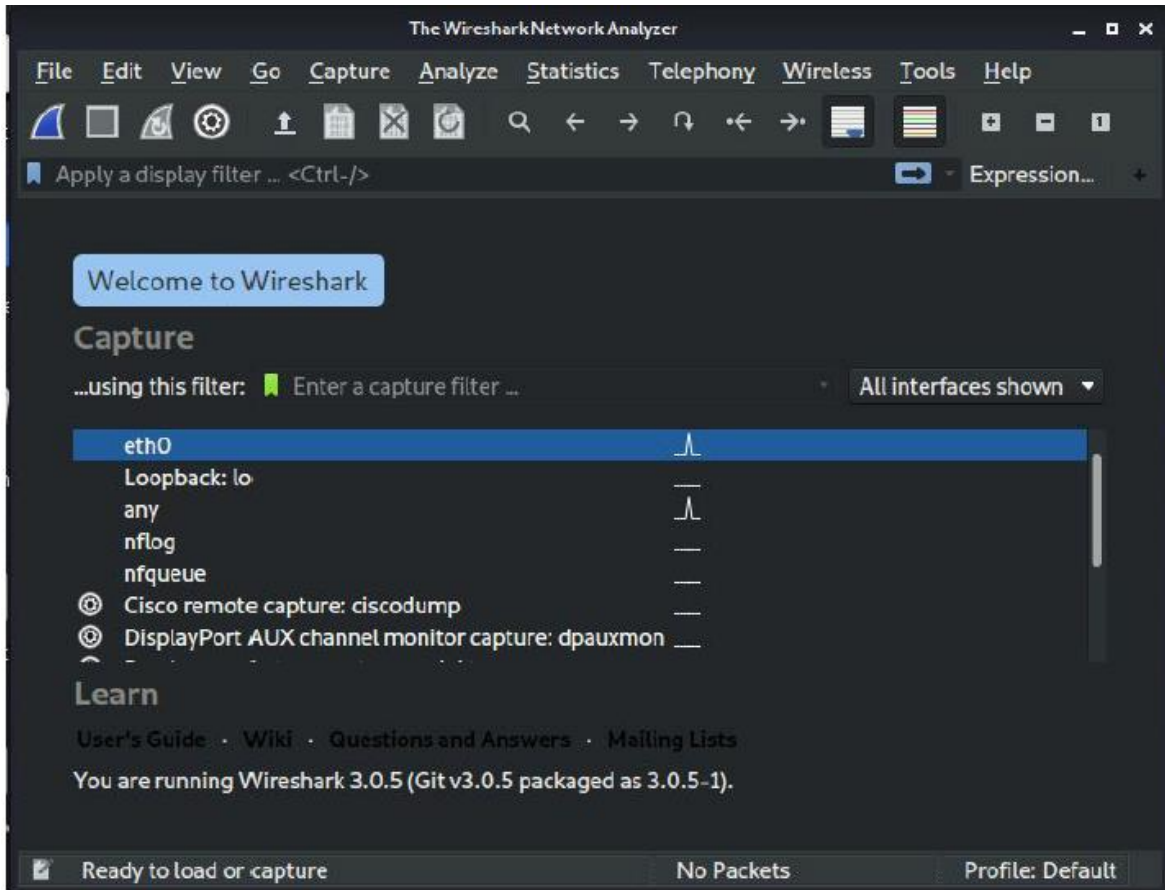
https://instagram.com/the_guy_that_hacks

Traducción por: Carla Cortés y Luis Rodríguez

ÍNDICE

1. Cómo empezar
2. Empezar el análisis: Realizar una captura en vivo del tráfico de red/tráfico web
 - 2.1 2.1 Filtrar los paquetes con la barra de filtros durante la captura y explicar todos los posibles filtros utilizados por usted.
3. **Ver los resúmenes de los paquetes con la ventana de la lista de paquetes**
4. **Estudiar los detalles del paquete con la ventana de detalles del paquete**
5. **Ver los datos de los paquetes con la ventana de bytes de paquetes individuales**
6. **Simplemente navegar por Internet**
7. **Visualización de los datos de la cabecera del paquete**
 - 7.1 Captura de paquetes con Wireshark
 - 7.2. Explorar la capa de interfaz de red / capa de enlace de datos
 - 7.2.2. Ver datos de tramas Ethernet capturados con Wireshark
8. **Exploración de la capa de Internet**
 - 8.1.1 Cabecera IPv4: Imagen de abajo
 - 8.1.2. Ver los datos de la cabecera IP de un paquete TCP capturado con Wireshark
 - 8.1.3 Ver los datos de la cabecera IP de un paquete UDP
 - 8.1.4. Ver los datos de la cabecera IP de un paquete ARP
9. **Exploración de la capa de transporte**
 - 9.1.1. Cabecera TCP: Imagen de abajo
 - 9.1.2 Ver los datos de la cabecera TCP de un paquete TCP capturado con Wireshark
 - 9.1.3 Cabecera UDP: En la imagen de abajo
 - 9.1.4 Ver los datos de la cabecera UDP de un paquete UDP capturado con Wireshark
 - 9.1.5 Comparar y contrastar IP, TCP y UDP
10. 10. Explorar la capa de aplicación
 - 10.1.1 Analizar un paquete HTTP
 - 10.1.2 Analizar un paquete DNS
11. Preguntas comunes

1. Comenzando con Wireshark

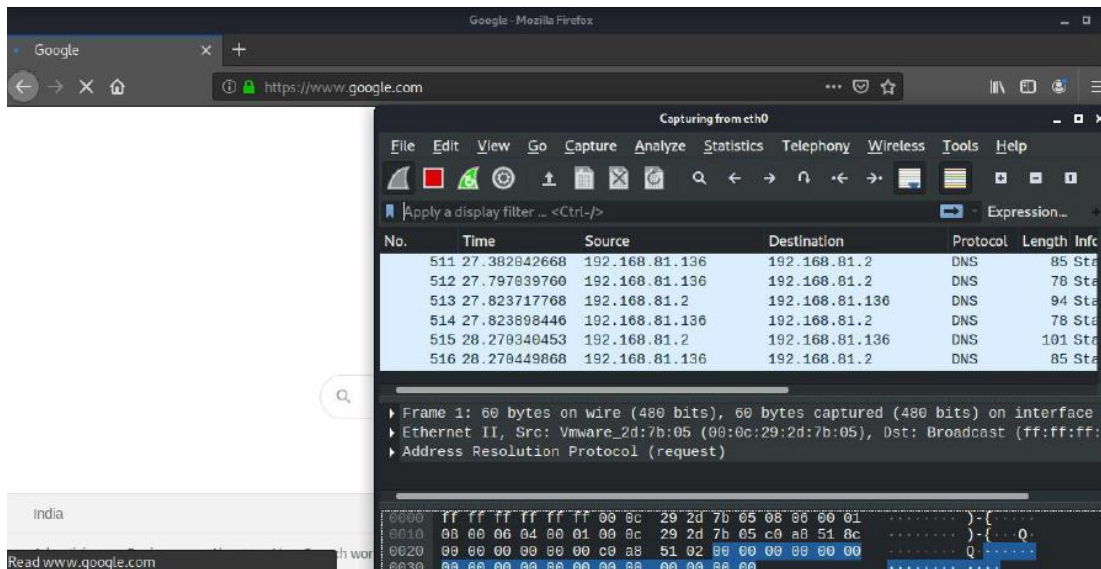


Wireshark está preinstalado en kali linux.

Wireshark es un analizador de paquetes gratuito y de código abierto. Se utiliza para la resolución de problemas de red, el análisis, el desarrollo de software y protocolos de comunicación, y la educación. Originalmente llamado Ethereal, el proyecto fue rebautizado como Wireshark en mayo de 2006 debido a problemas de marca.

La **GUI de wireshark** tiene

1. Barra de título
2. Menú principal
3. Barra de herramientas principal
4. Barra de herramientas del filtro
5. Lista de paquetes
6. Barra de desplazamiento inteligente
7. Detalles del paquete
8. Packet Bytes
9. Barra de estado



Haciendo clic en la interfaz eth0 se inicia la captura de paquetes, mientras que el **sniffing** podemos **analizar** y puede aplicar **filtros** para ver la necesidad exacta.

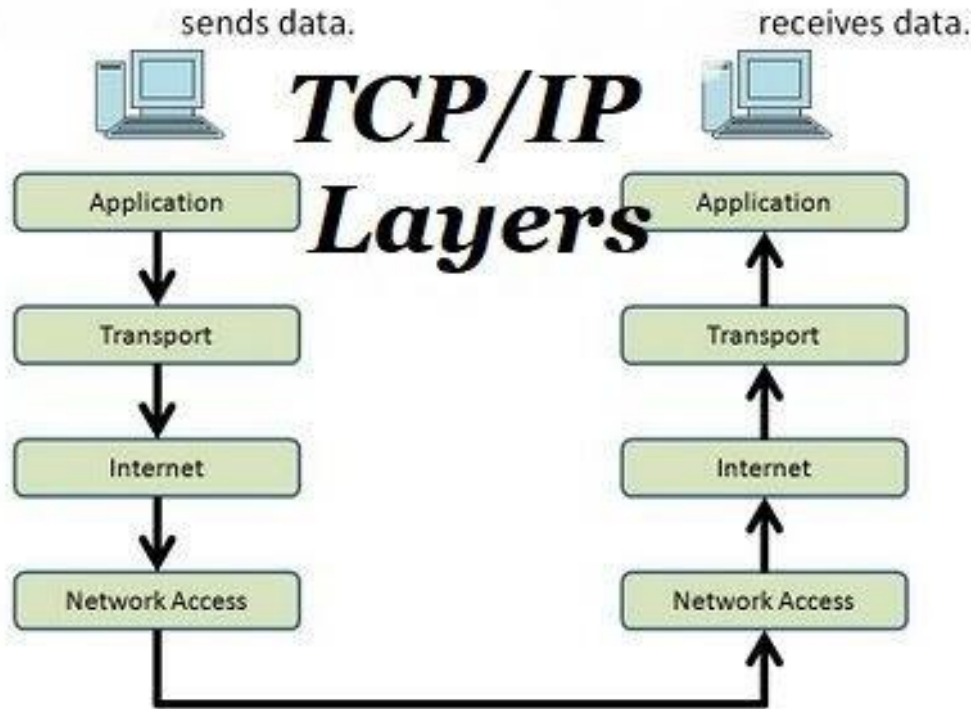
Definir las **cuatro capas del** modelo de referencia **TCP/IP**.

La capa TCP maneja el mensaje que se va a transmitir. Este mensaje se suele dividir en pequeñas unidades. Estas pequeñas unidades se conocen como paquetes. Además, estos paquetes se transmiten por la red.

Estos paquetes son recibidos por la capa TCP correspondiente en el receptor y reensamblados en el mensaje original.

El modelo TCP/IP tiene 4 capas, que son:

- Capa de aplicación
- Capa de transporte
- Capa de Internet
- Capa de red



Capa de aplicación:

La primera capa es la de aplicación. Esta capa proporciona a las aplicaciones un intercambio de datos estandarizado. A continuación se indican los protocolos de estas capas:

- Protocolo de transferencia de hipertexto (HTTP)
- Protocolo de transferencia de archivos (FTP)
- Protocolo de oficina de correos 3 (POP3)
- Protocolo simple de transferencia de correo (SMTP)
- Protocolo simple de gestión de redes (SNMP)

Este trabajo en capas con todos estos protocolos.

Capa de transporte:

La capa de transporte es la segunda capa del modelo TCP/IP. El trabajo básico de la capa de transporte es mantener las comunicaciones de extremo a extremo. A continuación se indican los protocolos de estas capas:

- TCP
- Protocolo de Datagramas de Usuario (UDP)

Estos dos protocolos se utilizan para la capa de transporte en TCP/IP.

Capa de red:

La tercera capa de TCP/IP es una capa de red. También se conoce como capa de Internet. La capa de red se ocupa de los paquetes. Los siguientes son los protocolos que se utilizan en esta capa.

- IP
- Protocolo de mensajes de control de Internet (ICMP) Capa física

La última capa es la capa física. Esta capa trabaja con los siguientes protocolos.

- Ethernet para LAN (redes de área local)
- Protocolo de resolución de direcciones (ARP)

Examinar **los datos de la cabecera del paquete** con Wireshark

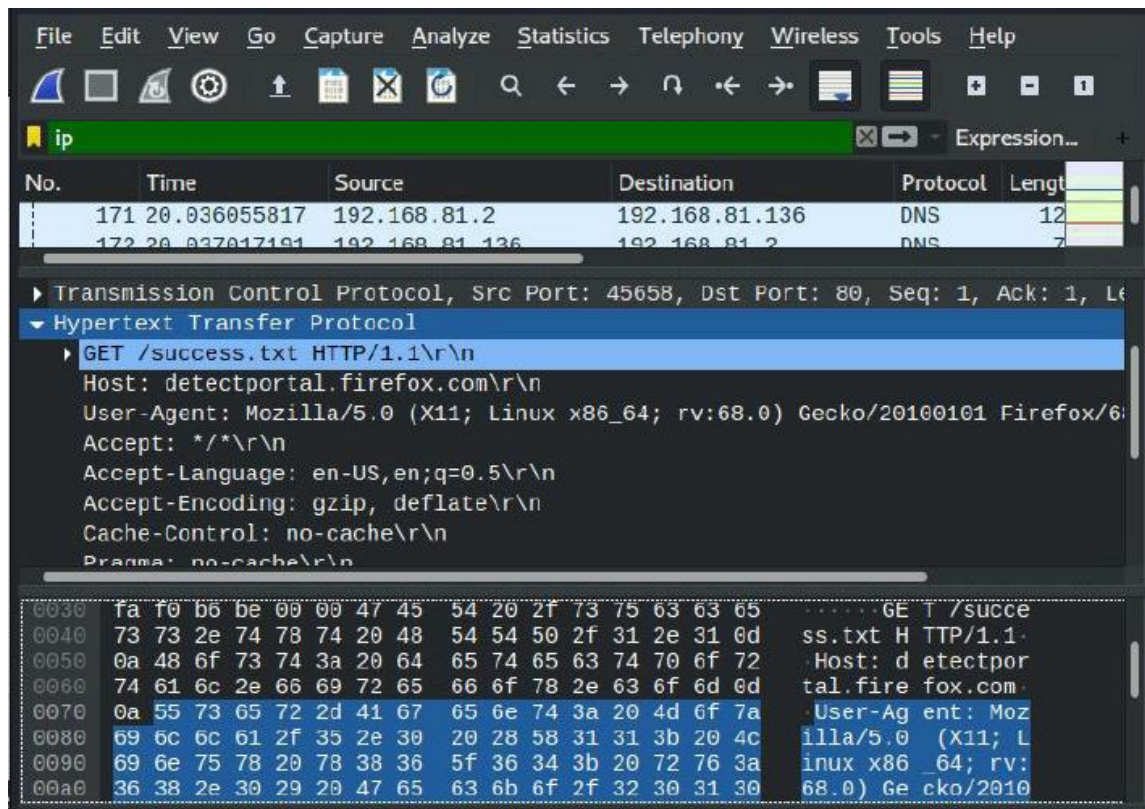
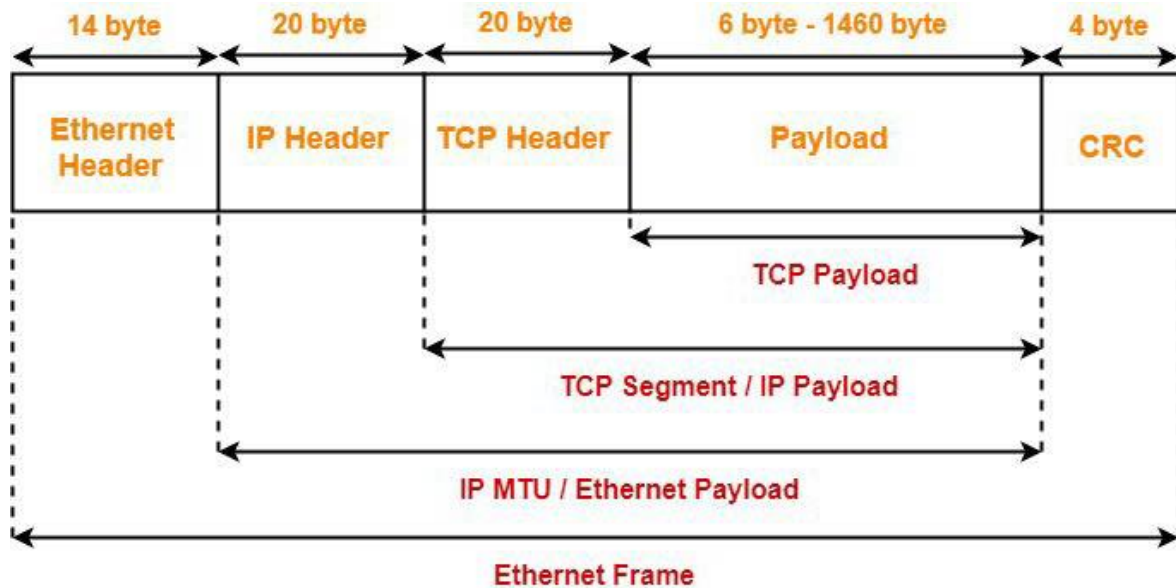


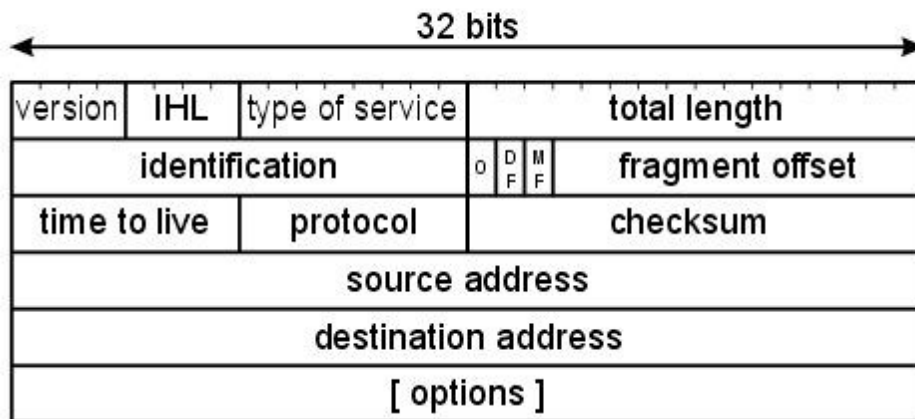
Imagen. Encabezados de los paquetes de datos mostrados arriba

Definir los **campos de cabecera** de las **tramas Ethernet**, del **Protocolo de Internet (IP)**, del **Protocolo de Control de Transporte (TCP)** y del **Protocolo de Datagramas de Usuario (UDP)** / diferentes tipos de cabeceras de paquetes, incluyendo los campos de cabecera y sus valores.



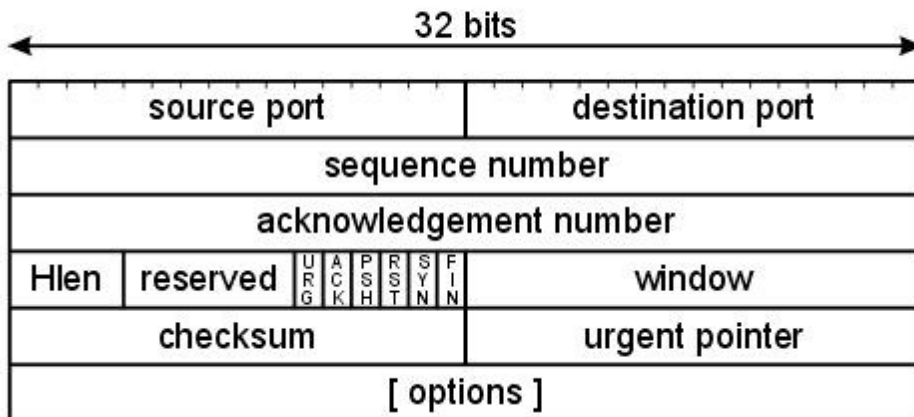
Ethernet

IP header format



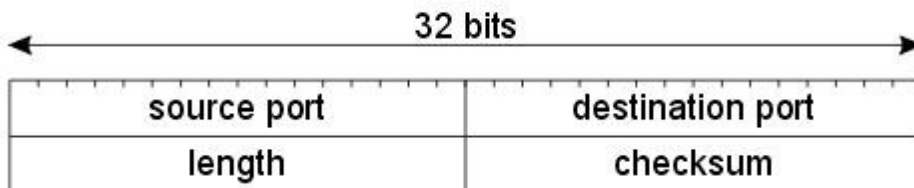
Protocolo IP

TCP header format



Cabecera TCP

UDP header format



Cabecera UDP

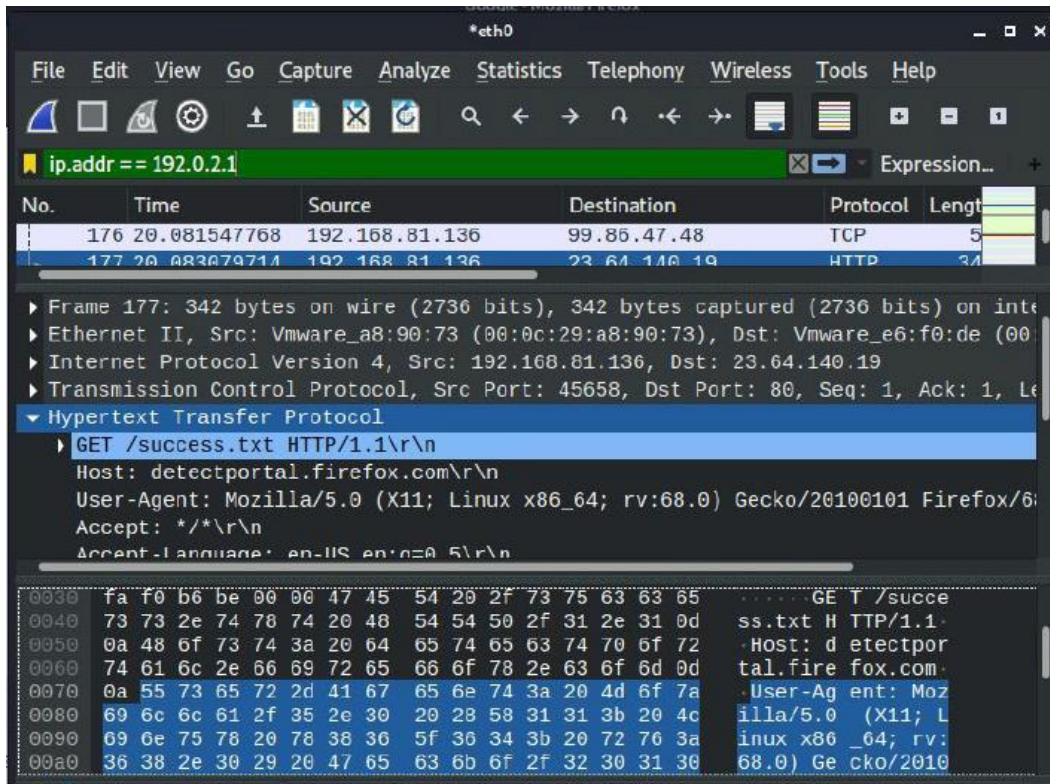
Compare y contraste **TCP** y **UDP**.

Differences are-

| Properties | TCP | UDP |
|-------------------|-------------------------------|--------------------------------------|
| Header | Dynamic header (20 – 60 B) | Static header of 8 Bytes |
| Max segment | any size or 2^{30} B | short message 65536 Bytes |
| Flow Control | Yes, Window and seq. no . | NO |
| Checksum | Compulsory | Optional |
| Connection nature | TCP+ IP = connection oriented | UDP+ IP= connection less |
| Error control | Own mechanism | Depends on ICMP (No self feature) |
| Support multicast | NO | YES |
| Support broadcast | NO | Yes |
| Examples service | HTTP,SMTP,FTP,TELNET | TFTP,DNS,SNMP |

2. Empezar el análisis: Realice una captura en vivo del tráfico de red/tráfico web

2.1 Filtrar los paquetes con la barra de filtros durante la captura y explicar todos los posibles filtros utilizados por usted.



Capturar sólo el tráfico hacia o desde la dirección IP
172.18.5.4: host 172.18.5.4

Capturar el tráfico hacia o desde un rango de direcciones IP:
192.168.0.0/24 o 192.168.0.0 máscara 255.255.255.0

Capturar el tráfico de un rango de direcciones IP:
192.168.0.0/24 o 192.168.0.0 máscara de red 255.255.255.0

Capturar el tráfico hacia un rango de direcciones IP:
192.168.0.0/24 o 192.168.0.0 máscara de red 255.255.255.0

Capturar sólo el tráfico DNS (puerto 53): puerto 53

Capture el tráfico no HTTP y no SMTP en su servidor (ambos son equivalentes):

host www.example.com y no (puerto 80 o puerto 25)
host www.example.com y no puerto 80 y no puerto 25

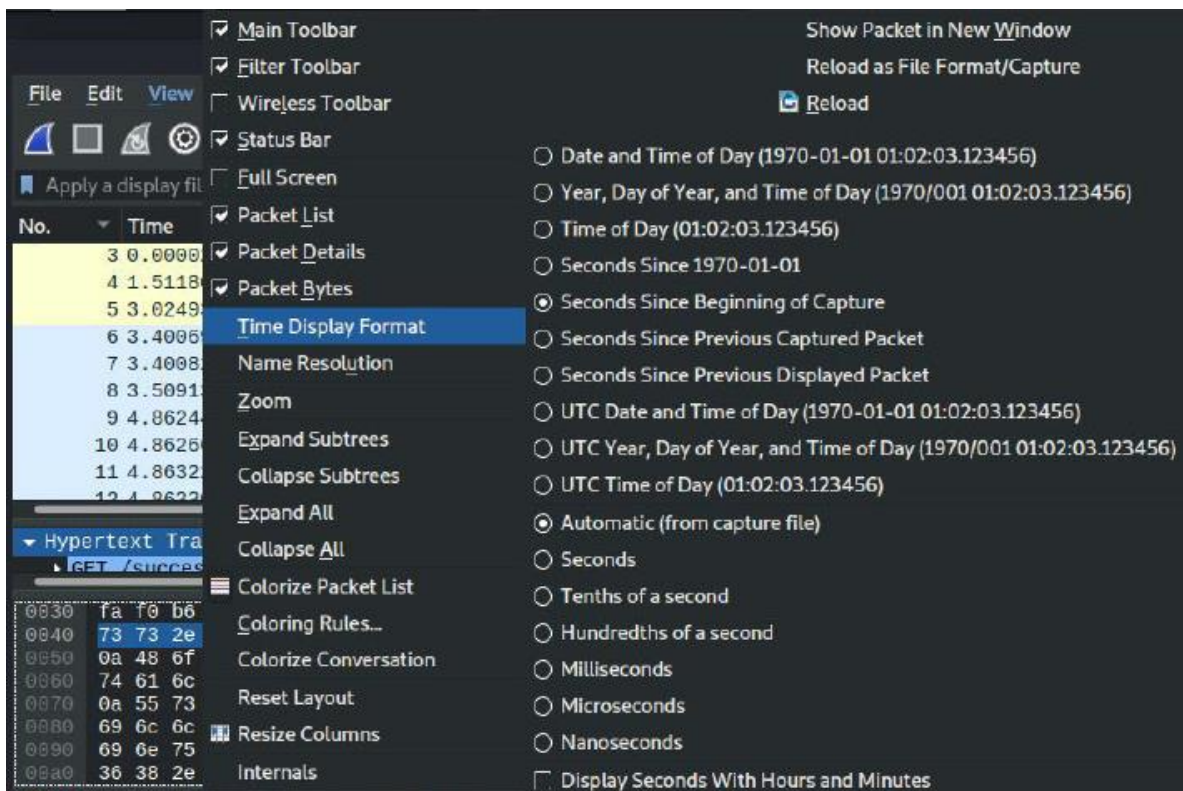
Captura excepto todo el tráfico ARP y DNS: puerto no 53 y no arp

Para capturar el tráfico de la vlan

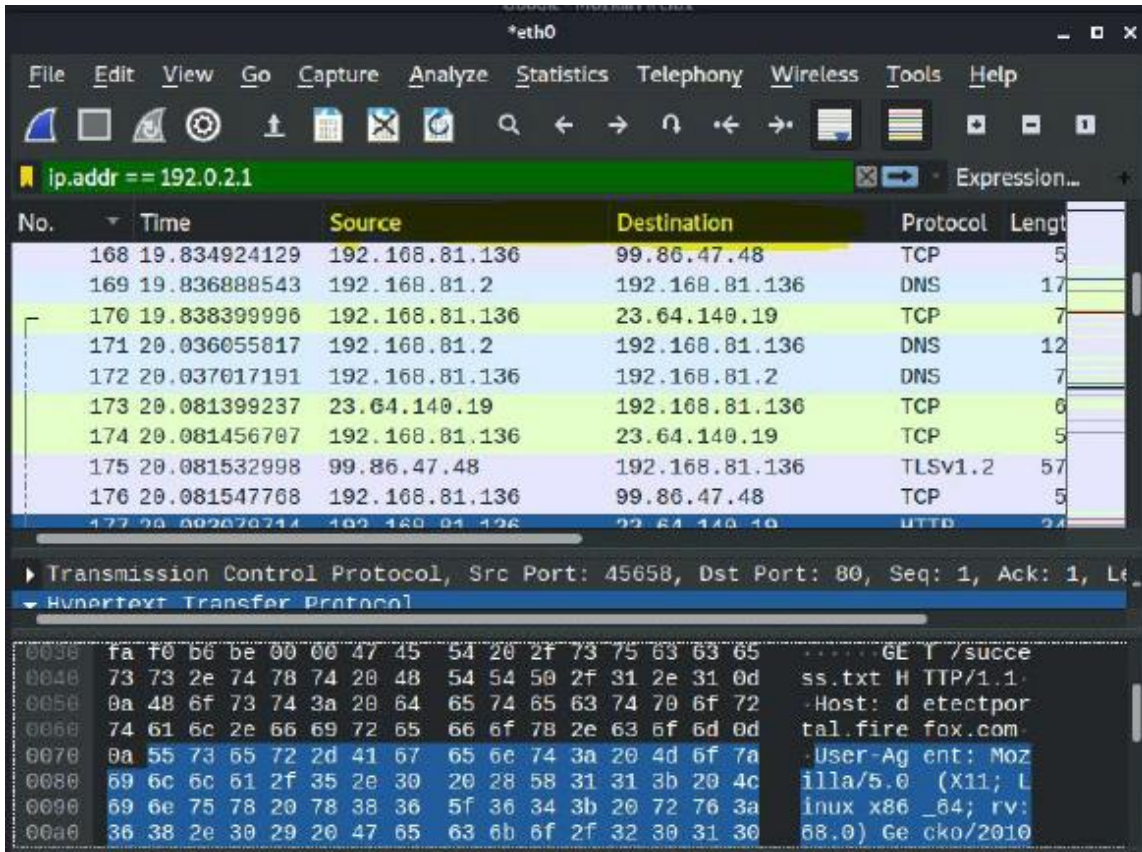
3. Ver los resúmenes de los paquetes con la ventana de la lista de paquetes

| No. | Time | Source | Destination | Protocol | Length |
|-----|-------------|----------------|----------------|----------|--------|
| 3 | 0.000020180 | 192.168.81.140 | 192.168.81.2 | NBNS | 11 |
| 4 | 1.511808997 | 192.168.81.140 | 192.168.81.2 | NBNS | 11 |
| 5 | 3.024931189 | 192.168.81.140 | 192.168.81.2 | NBNS | 11 |
| 6 | 3.400696460 | 192.168.81.136 | 192.168.81.2 | DNS | 8 |
| 7 | 3.400823404 | 192.168.81.136 | 192.168.81.2 | DNS | 8 |
| 8 | 3.509137922 | 192.168.81.2 | 192.168.81.136 | DNS | 24 |
| 9 | 4.862447872 | 192.168.81.136 | 192.168.81.2 | DNS | 7 |
| 10 | 4.862660426 | 192.168.81.136 | 192.168.81.2 | DNS | 7 |
| 11 | 4.863220065 | 192.168.81.136 | 192.168.81.2 | DNS | 7 |
| 12 | 4.863306621 | 192.168.81.136 | 192.168.81.2 | DNS | 7 |

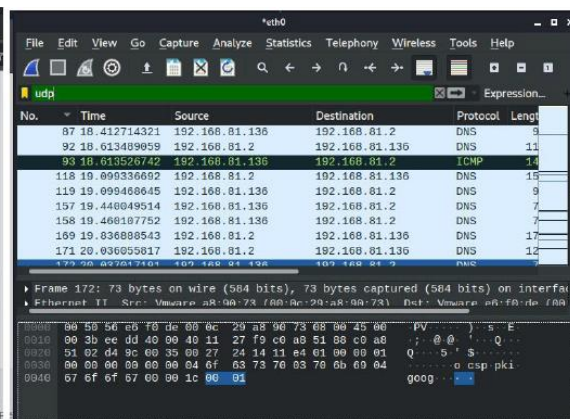
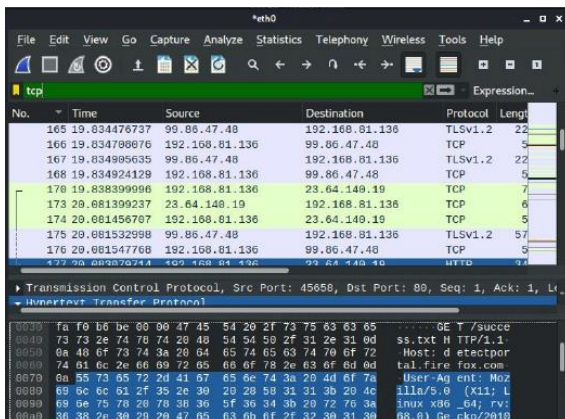
Número de paquete (No.): Los números de cada paquete comienzan con 1 para el primer paquete.



Timestamp (Time): Por defecto es el número de segundos desde el inicio de la captura



Direcciones IP (Origen: Source, Destino: Destination): La dirección de origen y destino del paquete.



Protocolos (Protocol) : El protocolo de paquetes (TCP, UDP, NBNS, etc.).

The screenshot shows the Wireshark interface with the filter 'tcp.flags.syn' applied. The packet list pane displays several packets, with packet 170 selected. The packet details pane shows the structure of the selected packet, including Ethernet II, Internet Protocol Version 4, and Transmission Control Protocol (TCP) fields. The TCP field is expanded to show the flags section, where the SYN flag is highlighted in blue.

| No. | Time | Source | Destination | Protocol | Length |
|-----|--------------|----------------|----------------|----------|--------|
| 160 | 19.628422937 | 192.168.81.136 | 216.58.196.106 | TCP | 5 |
| 161 | 19.628541353 | 216.58.196.106 | 192.168.81.136 | TCP | 129 |
| 162 | 19.628562008 | 192.168.81.136 | 216.58.196.106 | TCP | 5 |
| 163 | 19.628628668 | 216.58.196.106 | 192.168.81.136 | TLSv1.3 | 64 |
| 164 | 19.628644409 | 192.168.81.136 | 216.58.196.106 | TCP | 5 |
| 165 | 19.834476737 | 99.86.47.48 | 192.168.81.136 | TLSv1.2 | 22 |
| 166 | 19.834708076 | 192.168.81.136 | 99.86.47.48 | TCP | 5 |
| 167 | 19.834905635 | 99.86.47.48 | 192.168.81.136 | TLSv1.2 | 22 |
| 168 | 19.834924129 | 192.168.81.136 | 99.86.47.48 | TCP | 5 |
| 170 | 19.838300006 | 192.168.81.136 | 216.58.196.106 | TCP | 7 |

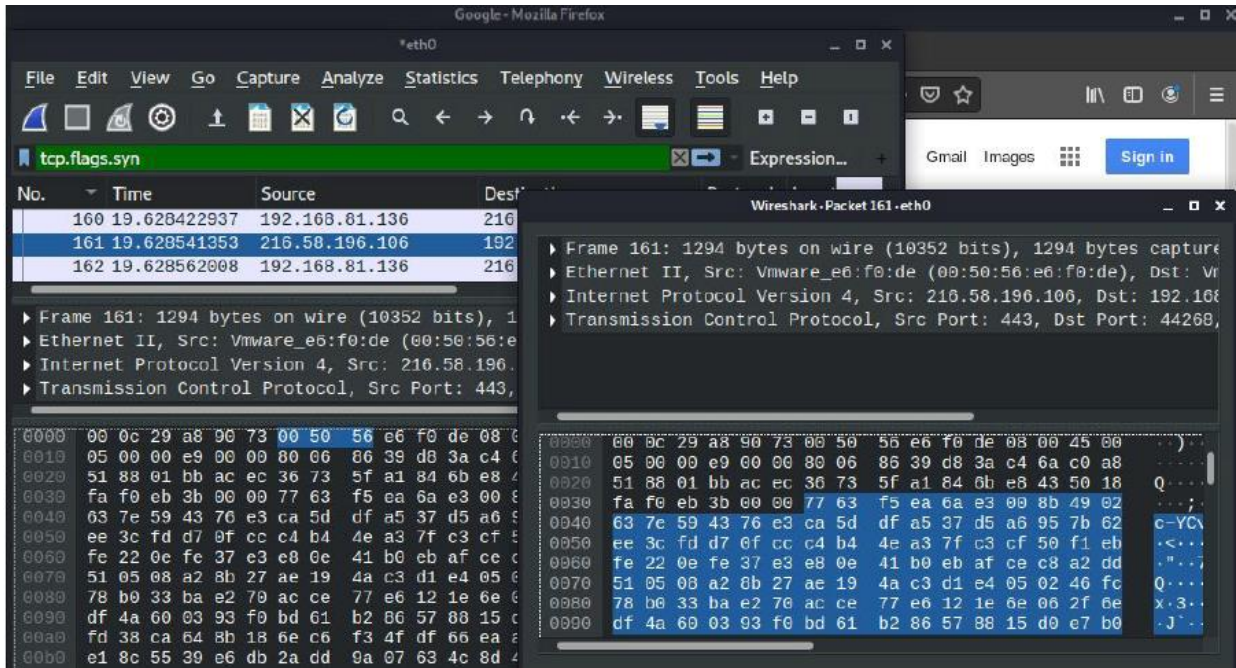
Frame 170: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface
 Ethernet II Src: VMware a8:90:73 (00:0c:29:a8:90:73) Dst: VMware e6:f0:de (00:0c:29:e6:f0:de)

```

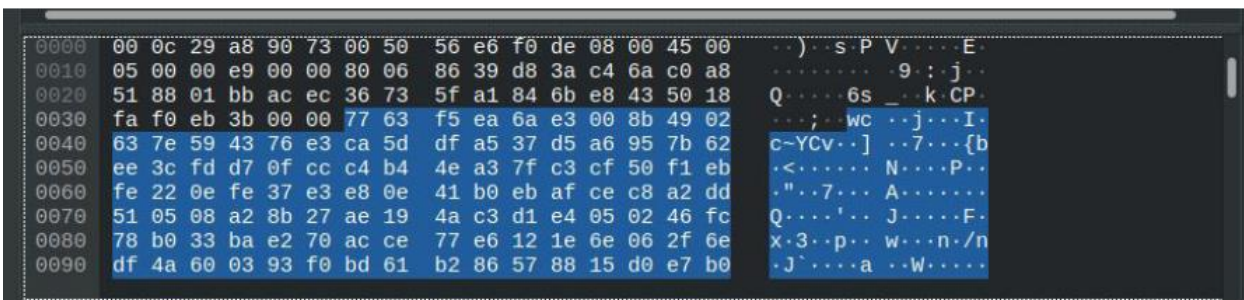
0000  00 50 56 e6 f0 de 00 0c 29 a8 90 73 08 00 45 00  -PV.....)..S.E
0010  00 3c a0 35 40 00 40 06 e5 02 c0 a8 51 88 17 40  -<.5@.@.....Q..@
0020  8c 13 b2 5a 00 50 03 b7 44 f5 00 00 00 00 a0 02  -..Z.P..D.....
0030  fa f0 b5 b2 00 00 02 04 05 b4 04 02 08 0a 44 c0  -.....D.....
0040  48 d9 00 00 00 00 01 03 03 07                    H.....
  
```

Información adicional del protocolo (info): Ejemplo: para un paquete TCP, este campo indica si es un paquete SYN, ACK o FIN.

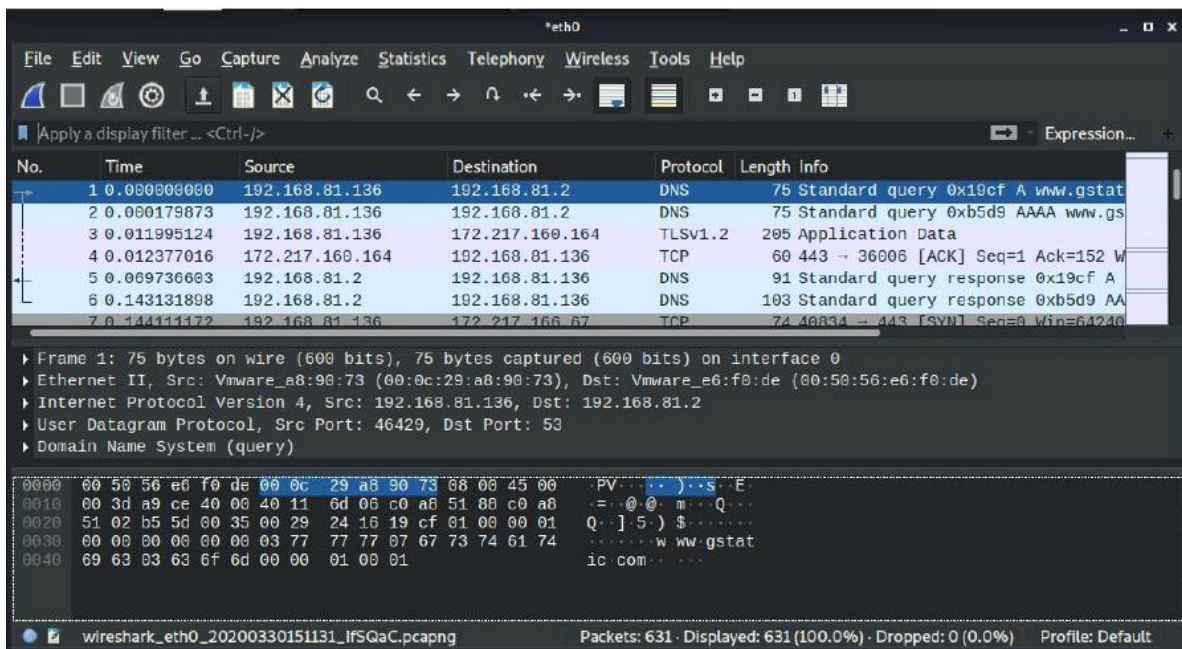
4. Estudiar los detalles del paquete con la ventana de detalles del paquete



5. Ver los datos de los paquetes con la ventana de bytes de paquetes individuales

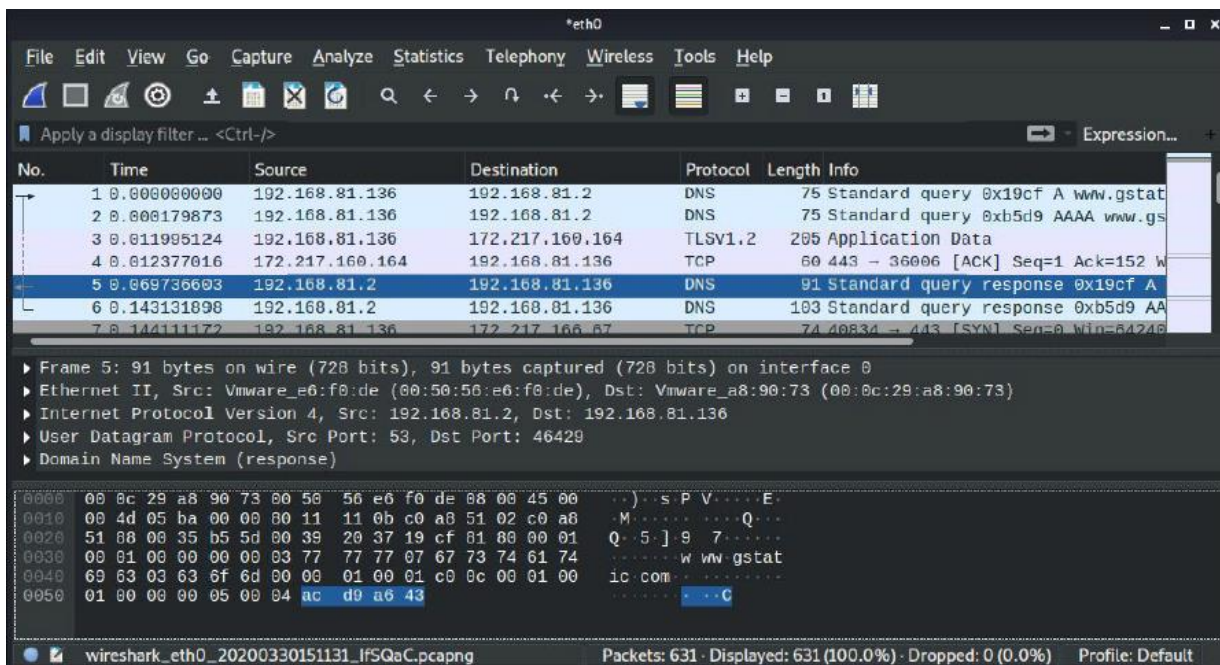


6. Simplemente navegar por Internet



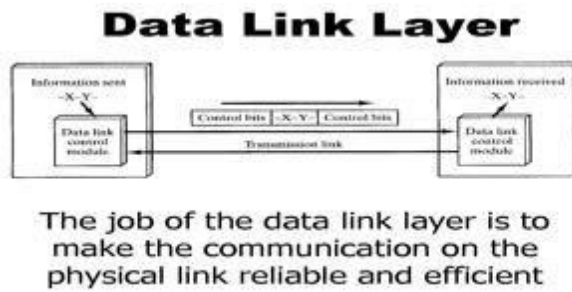
Datos después de navegar por Internet

7. Visualización de los datos de la cabecera del paquete

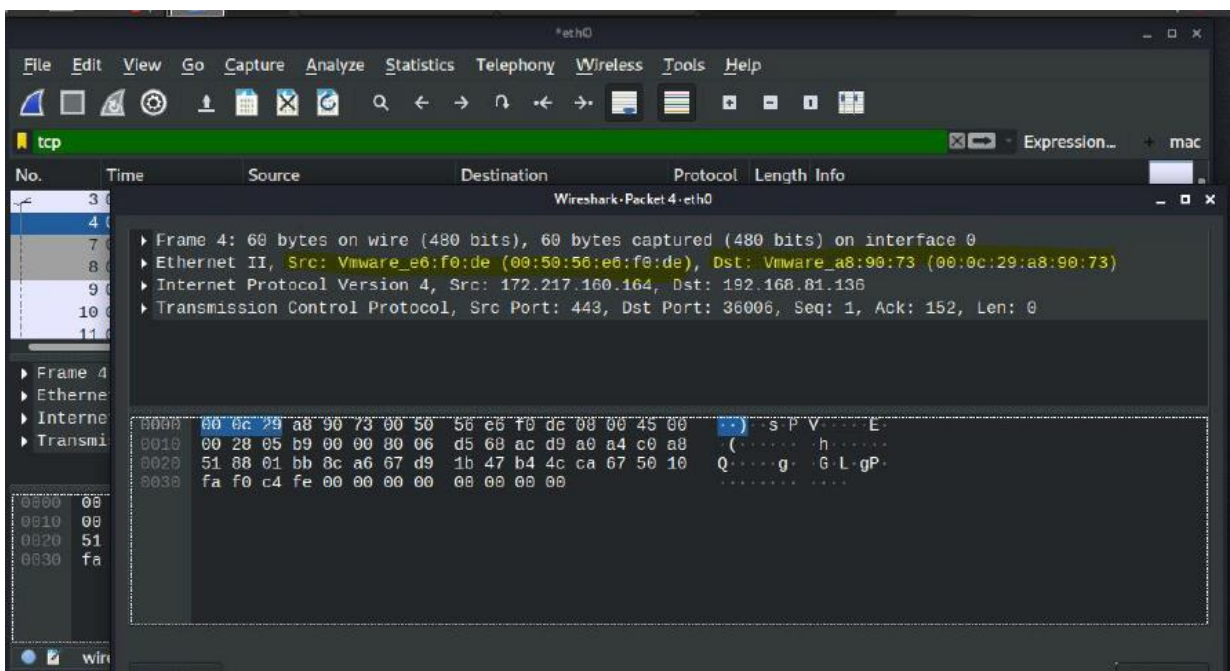


7.1 Captura de paquetes con Wireshark

7.2. Explorar la capa de interfaz de red / capa de enlace de datos



7.2.2. Ver datos de tramas Ethernet capturados con Wireshark



8.1 Exploración de la capa de Internet

8.1.1. Cabecera IPv4: En la imagen de abajo

| | | | | |
|--|----------|-----------------|-----------------|-----------------|
| Version = 4 | HL | Type Of service | Total Length | |
| Identification | | | Flag | Fragment offset |
| Time to Live | Protocol | | Header Checksum | |
| Home Address : home agent address 130.45.10.20/16 | | | | |
| Destination Address : 14.56.8.9/8 | | | | |
| Protocol | S | Reserved | Header Checksum | |
| Destination Address mobile host home address 130.45.6.7/16 | | | | |
| Source Address (remote host) 200.4.7.14/24 | | | | |
| Payload | | | | |

8.1.2. Ver los datos de la cabecera IP de un paquete TCP capturado con Wireshark

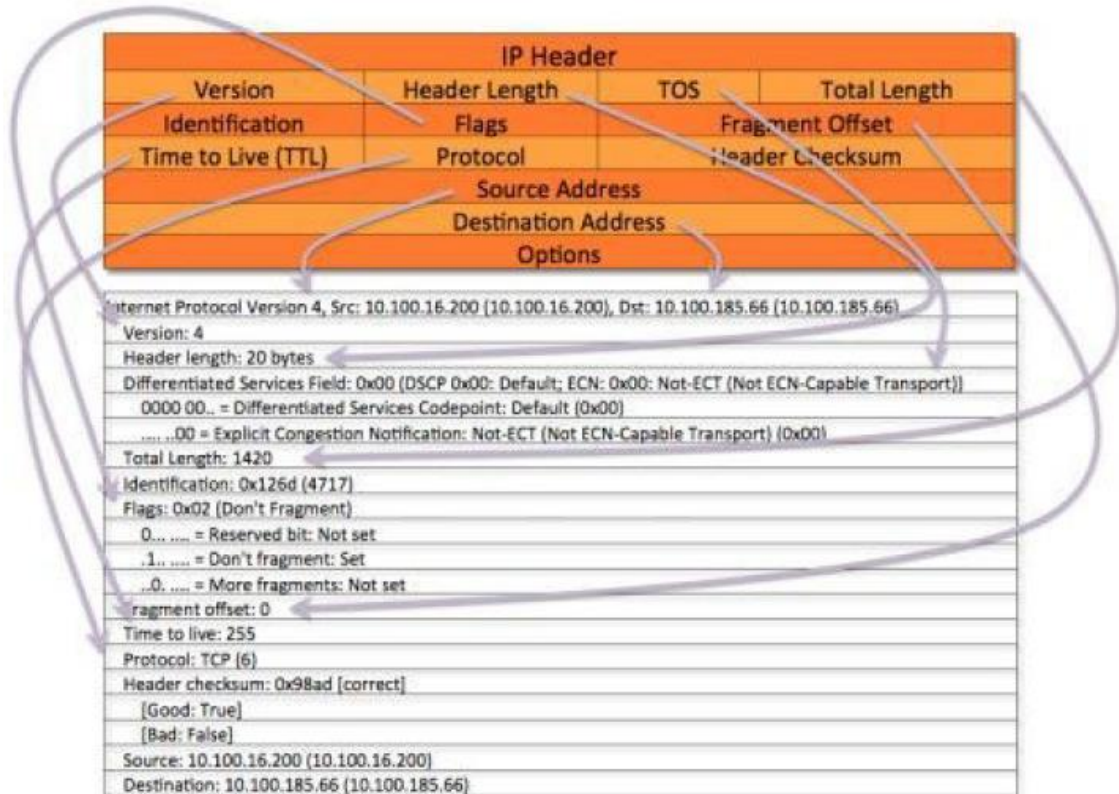
| TCP Header | | | |
|---|----------|----------------------------|-------------|
| Source Port Number | | Destination Port Number | |
| Sequence Number | | | |
| Acknowledgement Number | | | |
| Data Offset | Reserved | Flags ACK URG RST SYN etc. | Window Size |
| Checksum | | Urgent Pointers | |
| Transmission Control Protocol, Src Port: 55075 (55075), Dst Port: 50100 (50100), Seq: 1381, Ack: 1, Len: 1380 | | | |
| Source port: 55075 (55075) | | | |
| Destination port: 50100 (50100) | | | |
| [Stream index: 10] | | | |
| Sequence number: 1381 (relative sequence number) | | | |
| [Next sequence number: 2761 (relative sequence number)] | | | |
| Acknowledgement number: 1 (relative ack number) | | | |
| Header length: 20 bytes | | | |
| Flags: 0x10 (ACK) | | | |
| 000... = Reserved: Not set | | | |
| ...0... = Nonce: Not set | | | |
| ...0... = Congestion Window Reduced (CWR): Not set | | | |
| ...0... = ECN-Echo: Not set | | | |
| ...0... = Urgent: Not set | | | |
| ...1... = Acknowledgement: Set | | | |
| ...0... = Push: Not set | | | |
| ...0... = Reset: Not set | | | |
| ...0... = Syn: Not set | | | |
| ...0... = Fin: Not set | | | |
| Window size value: 4380 | | | |
| [Calculated window size: 4380] | | | |
| [Window size scaling factor: 1] | | | |
| Checksum: 0x18 [validation disabled] | | | |
| [Good Checksum: False] | | | |
| [Bad Checksum: False] | | | |
| [SEQ/ACK analysis] | | | |
| [Bytes in flight: 2760] | | | |
| Data (1380 bytes) | | | |

8.1.3 Ver los datos de la cabecera IP de un paquete UDP

```

▶ Frame 1: 75 bytes on wire (600 bits), 75 bytes captured (600 bits) on interface 0
▼ Ethernet II, Src: Vmware_a8:90:73 (00:0c:29:a8:90:73), Dst: Vmware_e6:f0:de (00:50:56:e6:f0:de)
  ▶ Destination: Vmware_e6:f0:de (00:50:56:e6:f0:de)
  ▶ Source: Vmware_a8:90:73 (00:0c:29:a8:90:73)
  Type: IPv4 (0x0800)
▼ Internet Protocol Version 4, Src: 192.168.81.136, Dst: 192.168.81.2
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
  ▶ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 61
    Identification: 0xa9ce (43470)
  ▶ Flags: 0x4000, Don't fragment
    Time to live: 64
    Protocol: UDP (17)
    Header checksum: 0x6d06 [validation disabled]
    [Header checksum status: Unverified]
    Source: 192.168.81.136
    Destination: 192.168.81.2
0000  00 50 56 e6 f0 de 00 0c 29 a8 90 73 08 00 45 00  .PV....)S.E
0010  00 3d a9 ce 40 00 40 11 6d 06 c0 a8 51 88 c0 a8  .-.@.m.Q.
0020  51 02 b5 5d 00 35 00 29 24 16 19 cf 01 00 00 01  Q.]5)S.....
0030  00 00 00 00 00 00 03 77 77 77 07 67 73 74 61 74  .....w ww.gstat
0040  69 63 63 63 6f 6d 00 00 01 00 01                ic.com. ....
  
```

8.1.4. Ver los datos de la cabecera IP de un paquete ARP

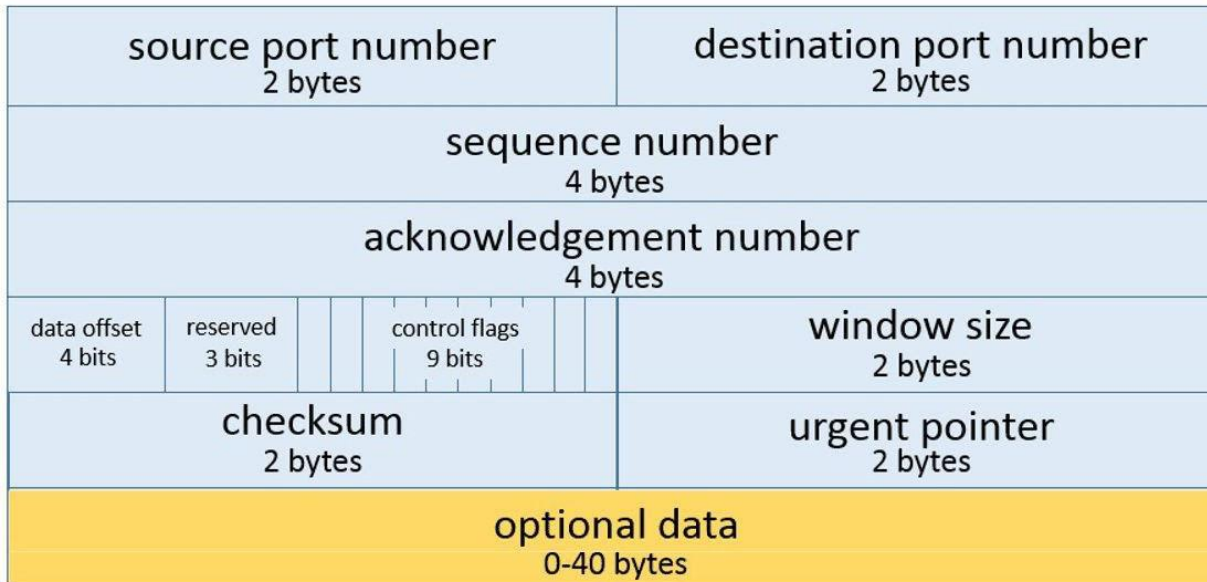


9 Exploración de la capa de transporte

9.1.1. Cabecera TCP: Imagen de abajo

Transmission Control Protocol (TCP) Header

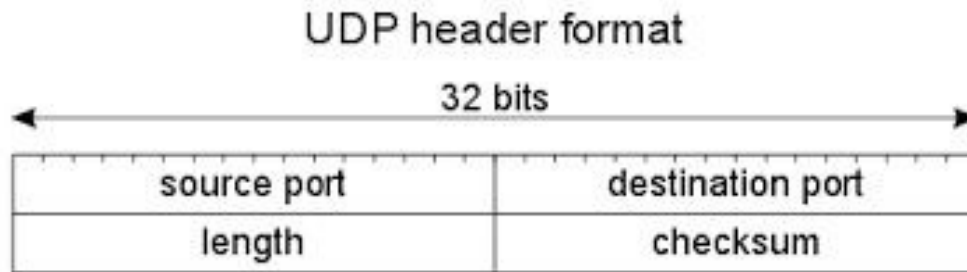
20-60 bytes



9.1.2 Ver los datos de la cabecera TCP de un paquete TCP capturado con Wireshark

```
Wireshark - Packet 4 - eth0
Ethernet II, Src: VMware_e6:f0:de (00:50:56:e6:f0:de), Dst: VMware_a8:90:73 (00:0c:29:a8:90:73)
Internet Protocol Version 4, Src: 172.217.160.164, Dst: 192.168.81.136
  0100 ... = Version: 4
  ... 0101 = Header Length: 20 bytes (5)
  Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 40
    Identification: 0x05b9 (1465)
  Flags: 0x0000
    Time to live: 128
    Protocol: TCP (6)
    Header checksum: 0xd568 [validation disabled]
    [Header checksum status: Unverified]
    Source: 172.217.160.164
    Destination: 192.168.81.136
  Transmission Control Protocol, Src Port: 443, Dst Port: 36086, Seq: 1, Ack: 152, Len: 0
    0000  00 0c 29 a8 90 73 00 50 56 e6 f0 de 08 00 45 00  ...s.PV....E.
    0010  00 28 05 b9 00 00 00 06 d5 68 ac d9 a0 a4 c0 a8  ..(.....h....
    0020  51 88 01 bb 8c a6 67 d9 1b 47 b4 4c ca 67 50 10  Q.....g..G.L.gP.
    0030  fa f0 c4 fe 00 00 00 00 00 00 00 00  .....
```

9.1.3 Cabecera UDP: En la imagen de abajo



9.1.4 Ver los datos de la cabecera UDP de un paquete UDP capturado con Wireshark

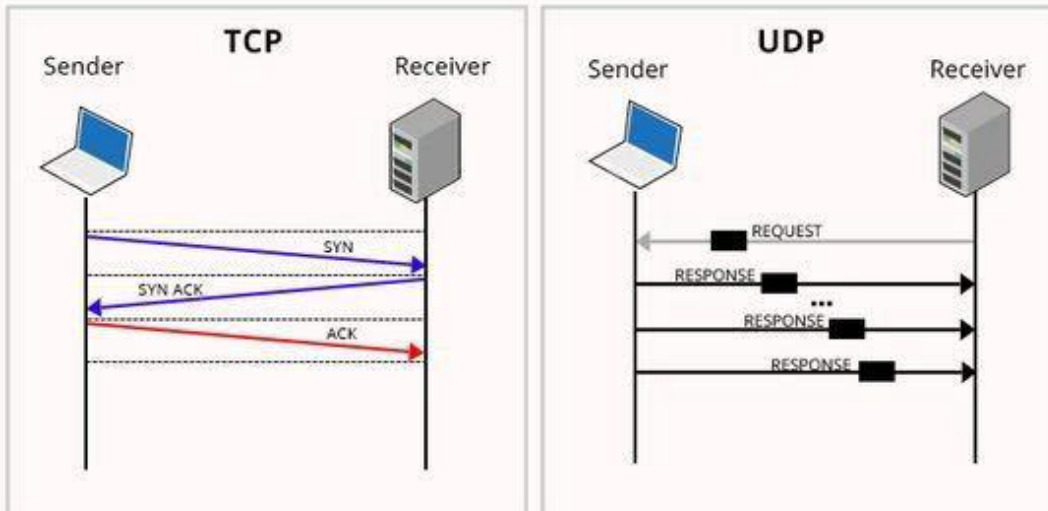
```
▶ Frame 1: 75 bytes on wire (600 bits), 75 bytes captured (600 bits) on interface 0
▼ Ethernet II, Src: Vmware_a8:90:73 (00:0c:29:a8:90:73), Dst: Vmware_e6:f0:de (00:50:56:e6:f0:de)
  ▶ Destination: Vmware_e6:f0:de (00:50:56:e6:f0:de)
  ▶ Source: Vmware_a8:90:73 (00:0c:29:a8:90:73)
  Type: IPv4 (0x0800)
▶ Internet Protocol Version 4, Src: 192.168.81.136, Dst: 192.168.81.2
▼ User Datagram Protocol, Src Port: 46429, Dst Port: 53
  Source Port: 46429
  Destination Port: 53
  Length: 41
  Checksum: 0x2416 [unverified]
  [Checksum Status: Unverified]
  [Stream index: 0]
  ▶ [TimeStamps]
▶ Domain Name System (query)
```



```
0016  00 3d a9 ce 40 00 40 11 6d 06 c0 a8 51 88 c0 a8  = ..@.m...Q..
0020  51 02 b5 5d 00 35 00 29 24 16 19 cf 01 00 00 01  Q.]5.)$. . . . .
```


9.1.5 Comparar y contrastar IP, TCP y UDP

TCP Vs UDP Communication



10. Explorar la capa de aplicación

10.1.1 Analizar un paquete HTTP

```
▶ Frame 592: 428 bytes on wire (3424 bits), 428 bytes captured (3424 bits) on interface 0
▼ Ethernet II, Src: Vmware_a8:90:73 (00:0c:29:a8:90:73), Dst: Vmware_e6:f0:de (00:50:56:e6:f0:de)
  ▶ Destination: Vmware_e6:f0:de (00:50:56:e6:f0:de)
  ▶ Source: Vmware_a8:90:73 (00:0c:29:a8:90:73)
  Type: IPv4 (0x0800)
▶ Internet Protocol Version 4, Src: 192.168.81.136, Dst: 172.217.174.227
▶ Transmission Control Protocol, Src Port: 43902, Dst Port: 80, Seq: 1, Ack: 1, Len: 374
▼ Hypertext Transfer Protocol
  ▶ POST /gts1o1 HTTP/1.1\r\n
  Host: ojsp.pki.goog\r\n
  User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0\r\n
  Accept: */*\r\n
  Accept-Language: en-US,en;q=0.5\r\n
  Accept-Encoding: gzip, deflate\r\n
  Content-Type: application/ocsp-request\r\n
  Content-Length: 84\r\n
  Connection: keep-alive\r\n
  \r\n
  [url] [http://ocsp.pki.goog/gts1o1]
  [HTTP request 1/1]
  [Application/javascript]
0000  08 50 56 e6 f0 de 00 0c 29 a8 90 73 08 00 45 00  PV.... ) s . E
0010  81 90 2b 5d 40 00 40 06 a0 0f c9 a8 51 88 ac d9  ++]@.@ . . Q .
0020  ae e3 ab 7e 00 50 2a 9b ff 54 6b 8a 56 8b 50 18  ...~P*. .Tk V P.
```

10.1.2 Analizar un paquete DNS

The screenshot shows the Wireshark interface with a packet list table and a detailed packet view for a DNS query response.

| No. | Time | Source | Destination | Protocol | Length | Info |
|-----|-------------|----------------|----------------|----------|--------|-----------------------------------|
| 575 | 8.511296437 | 192.168.81.2 | 192.168.81.136 | DNS | 155 | Standard query response 0x057e AA |
| 583 | 8.816888723 | 192.168.81.136 | 192.168.81.2 | DNS | 73 | Standard query 0x6d75 A ojsp.pki |

The detailed view for packet 575 shows:

- Frame 575: 155 bytes on wire (1240 bits), 155 bytes captured (1240 bits) on interface 0
- Ethernet II, Src: Vmware_e6:f0:de (00:50:56:e6:f0:de), Dst: Vmware_a8:90:73 (00:0c:29:a8:90:73)
- Destination: Vmware_a8:90:73 (00:0c:29:a8:90:73)
- Source: Vmware_e6:f0:de (00:50:56:e6:f0:de)
- Type: IPv4 (0x0800)
- Internet Protocol Version 4, Src: 192.168.81.2, Dst: 192.168.81.136
- User Datagram Protocol, Src Port: 53, Dst Port: 40513
- Source Port: 53
- Destination Port: 40513
- Length: 121
- Checksum: 0xd01e [unverified]
- [Checksum Status: Unverified]
- [Stream index: 8]

The hex dump at the bottom shows the raw bytes of the packet, with the IP address 192.168.81.2 highlighted in blue.

11. Preguntas comunes

Que. 1. ¿Captura Wireshark todo el tráfico de Internet? Si es así, explica por qué. Si no es así, ¿qué tráfico captura?

Respuesta: Lo más probable es que sólo vea el tráfico en el que participa su máquina, o que se difunde a todas las máquinas.

Esto se debe a que, durante años, la mayoría de las redes LAN se han construido sobre la base de la tecnología Ethernet conmutada, a diferencia de las redes basadas en concentradores o en buses. En esas tecnologías más antiguas, todos los equipos de la LAN veían todo el tráfico, simplemente porque todos estaban conectados eléctricamente entre sí. Con Ethernet conmutada, el conmutador decide qué paquetes enviar a cada puerto. Esto hace que la red sea más rápida y ligeramente más segura.

(La Ethernet conmutada no es una muy buena medida de seguridad, porque es fácil de derrotar con el envenenamiento ARP).

Ahora bien, es posible que todavía estés en una Ethernet basada en un hub, o algo similar. Eso sólo puede ocurrir con redes de 100 Mbit/s y más lentas. Parte de la especificación de Gigabit Ethernet es un requisito para los conmutadores. No encontrarás un concentrador GigE.

También hay que tener en cuenta que las redes inalámbricas se comportan efectivamente como las LAN de antaño: todos los equipos conectados a una determinada red Wi-Fi pueden ver todo el tráfico, debido puramente a la naturaleza de la comunicación por radio. Si estás en una LAN cableada con switches gestionados y tienes acceso administrativo a esos switches, probablemente encontrarás una característica que puedes activar en ellos llamada port mirroring. Esa característica existe específicamente para restaurar el antiguo comportamiento de la LAN preconmutada: designa un puerto como especial, dirigiendo copias de todo el tráfico hacia él, incluso los paquetes que no están dirigidos a las direcciones MAC conectadas a ese puerto.

Que. 2. Escriba los filtros de Wireshark para: Ver el tráfico UDP cuando se realiza la exploración.

Respuesta: simplemente escriba UDP y presione enter, y podrá ver todos los paquetes udp que fueron capturados.

Que. 3. Vea el tráfico ICMP de cualquier dirección.

Respuesta: Para analizar el tráfico ICMP Echo Request:

1. Observe el tráfico capturado en el panel superior de la lista de paquetes de Wireshark. Busque el tráfico con ICMP como protocolo. Para ver sólo el tráfico ICMP, escriba **icmp** (en minúsculas) en el cuadro Filtro y pulse **Intro**.
2. Seleccione el primer paquete ICMP, etiquetado como **Echo (ping) request**.
3. Observe los detalles del paquete en el panel central de detalles del paquete de Wireshark. Observe que se trata de una trama Ethernet II / Protocolo de Internet versión 4 / Protocolo de mensajes de control de Internet.
4. Expanda el Protocolo de Mensajes de Control de Internet para ver los detalles de ICMP.
5. Observe el tipo. Observe que el tipo es 8 (solicitud de eco (ping)).

6. Seleccione Datos en el panel central de detalles de paquetes de Wireshark para resaltar la parte de datos de la trama.
7. Observe el contenido del paquete en el panel inferior de bytes de paquetes de Wireshark. Observe que Windows envía una secuencia alfabética durante las solicitudes de ping.

Que. 4. ¿Por qué los paquetes ARP no tienen cabeceras IP?

Respuesta: Aunque hay direcciones IP o de protocolo utilizadas en este mensaje, en realidad no tiene una cabecera IP. Las direcciones IP que se ven son simplemente parte de la cabecera ARP. Esto significa que los mensajes ARP no son enrutables y que los routers no pasarán el tráfico ARP a otra red.

En consecuencia, no se puede determinar la dirección MAC de un nodo que no esté en la LAN del nodo de origen.

También significa que el Ethertype en una trama Ethernet que lleva un mensaje ARP es diferente que en el tráfico de datos estándar. Esta diferencia se muestra a continuación

```
⊕ Frame 17 (60 bytes on wire, 60 bytes captured)
⊖ Ethernet II, Src: Cisco_0d:18:57 (00:19:aa:0d:18:57), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
  ⊕ Destination: Broadcast (ff:ff:ff:ff:ff:ff)
  ⊕ Source: Cisco_0d:18:57 (00:19:aa:0d:18:57)
    Type: ARP (0x0806)
    Trailer: 0000000000000000000000000000000000000000
  ⊕ Address Resolution Protocol (request)
```

```
⊕ Frame 12 (74 bytes on wire, 74 bytes captured)
⊖ Ethernet II, Src: D-Link_c1:d2:01 (00:50:ba:c1:d2:01), Dst: Cisco_23:85:68 (00:19:06:23:85:68)
  ⊕ Destination: Cisco_23:85:68 (00:19:06:23:85:68)
  ⊕ Source: D-Link_c1:d2:01 (00:50:ba:c1:d2:01)
    Type: IP (0x0800)
  ⊕ Internet Protocol, Src: 192.168.10.11 (192.168.10.11), Dst: 129.21.21.1 (129.21.21.1)
  ⊕ Internet Control Message Protocol
```

Que. 5. Compare y contraste las cabeceras UDP y TCP.

Respuesta:

| Item | TCP | UDP |
|----------------------|---------------------------------------|---------------------------------------|
| Stands For | Transmission Control Protocol | User Datagram Protocol |
| Protocol | Connection Oriented | Connectionless |
| Security | Makes Checks For Errors And Reporting | Makes Error Checking But No Reporting |
| Data Sending | Slower | Faster |
| Header Size | 20 Bytes | 8 Bytes |
| Segments | Acknowledgement | No Acknowledgement |
| Typical Applications | - Email | - VoIP |

Que. 6. ¿Los paquetes ICMP especifican un puerto? Busca en Internet y explica por qué sí o por qué no.

Respuesta: **ICMP** es un protocolo diseñado específicamente para fines de diagnóstico y **el ping** no es más que una solicitud de eco ICMP y una respuesta de eco, por lo que no existe el concepto de números de puerto en **ICMP**. Los números de **puerto** son direcciones de la capa de transporte utilizadas por algunos protocolos de transporte.