Introducción a las redes II

PAULA DOLADO AYNIÉ

ICA0-M07U01I02 | 04-11-2022

Contenido

Actividad 1: Estudio de Protocolos

En un inicio explicaremos las líneas de los siguientes scripts que utilizaremos para estudiar una serie de tramas junto a los protocolos,

- Cmd

@echo off #El comando echo imprime por pantalla el texto que acompaña -> off.

Cls #Hace referencia a clear -> Limpia la pantalla.

Arp -d #Visualiza por pantalla la configuración del protocolo ARP.

Ipconfig /flushdns #Realiza una limpieza del DNS eliminando cualquier registro de caché.

Ping www.mytcpip.com -n2 -l300 #Determina el estado del host (web) -> Saber responde.

Hará un total de 2 respuestas y con su tamaño de 300 bytes especificados.

- Powershell

Get-NetNeighbor -AddressFamily ipv4 # Obtiene información de la caché de vecinos con una dirección IPv4.

Clear-DnsClientCache #Limpia la cache del DNS

Test-Connection www.mytcpip.com -count 3 -BufferSize 300 - #Hace una prueba de conexión para saber el estado del host (web). Buscará respuestas enviando 3 paquetes de igual tamaño de almacen datos(buffer) 300 bytes.

Actividad 2: Tramas

A continuación, se encontrarán capturas de tramas junto a sus respectivas explicaciones.

En primer lugar, utilizando el script de PowerShell debemos entender además de que hacen dichas ordenes, las acciones que realizaran en nuestro entorno del colegio.

TEORIA POWERSHELL:

De forma ordenada, nuestra maquina enviara a todas las maquinas que cuelgan de la red preguntando por la Mac del DNS, la cual le responderá con está obteniendo una segunda trama a diferencia de ser inicialmente una petición y seguidamente una respuesta. Paralelamente tienen en común ser tramas ARP debido a que se encarga de vincular la dirección MAC/dirección física a una dirección ip/lógica.

Seguidamente, al tener el DNS nuestra maquina inicial preguntara por la web que buscamos, proporcionándonos la dirección IP. Estas tramas serán llevadas a cabo por el protocolo DNS debido a que este se encarga de la relación entre dirección ip y nombre de dominio.

En nuestro caso de red la MAC del DNS coincide con la puerta de enlace por lo que nos "ahorraremos" dos tramas de petición y respuesta. Directamente enviara un conjunto de mensajes informativos mediante el protocolo encargado, ICMP, a la dirección ip de la web. Hasta que finalmente, la web nos responderá con un mensaje validando la conexión sin necesidad de hacer tantas peticiones ya que han quedado guardado en la caché mediante el protocolo ARP.

Dicho esto, en caso de que el DNS estuviera fuera como la primera trama que emitimos es un broadcast, a todos los dispositivos de la red, al no permanecer el DNS dentro de la red local nuestra maquina hará una petición apuntando la MAC del Gateway y esta después pedirá por DNS la web al servidor en cuestión hasta llegar a la web. Para la respuesta de la web nuestra maquina como vuelve a tener la ruta irá directamente al Gateway hasta nuestro pc que realiza el ping.

PRACTICA POWERSHELL:

Para observar las tramas que enviamos y recibimos deberemos iniciar el aplicativo Wireshark por el modo de red wifi debido a que nos encontramos ubicados en ella y ejecutaremos el script. Para no tener tramas en exceso de procesos que se estén realizando por peticiones de otros usuarios en la red detendremos la captura en cuanto se haya finalizado.

Para simplificar la búsqueda filtraremos las peticiones dependiendo del protocolo que lo realice, en un inicio, con el protocolo ARP y buscando por nuestra dirección IP de la maquina observaremos lo siguiente:

312	5.532123	VMware_5e:1f:87	Broadcast	ARP	60 Who has 10.0.3.116? Tell 10.0.0.2
313	5.532169	CyberTAN_72:3e:59	VMware_5e:1f:87	ARP	42 10.0.3.116 is at 00:45:e2:72:3e:59

De forma superficial podremos detectar la primera trama como bien dice broadcast preguntando q para que este lo tenga en su caché y en la segunda trama nos devuelve la MAC del DNS.

Si desglosamos dentro de la primera trama la información que nos proporciona el Wireshark por capas podemos ver que el primer desplegable nos muestra Frame 312 esto es su posición respecto al número de llegada, en este caso se refiere a la posición 312 además de otras indicaciones como la cantidad de bytes que se han capturado, con 480 bits enviamos datos.

Trama ARP: Análisis Ethernet

Sin extendernos en el apartado anterior es importante echar un ojo a la capa Ethernet (capa de enlace)[1]:

```
Ethernet II, Src: VMware_5e:1f:87 (00:0c:29:5e:1f:87), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
  > Destination: Broadcast (ff:ff:ff:ff:ff:ff)
  > Source: VMware_5e:1f:87 (00:0c:29:5e:1f:87)
    Type: ARP (0x0806)
    Padding: 000000000000000000000000000000000000
```

Como bien muestra una vista previa en el mismo desplegable podemos determinar el tipo de destino, en este caso es de broadcast. Dicha difusión se indica con 12 dígitos hexadecimales del 0-9 y a-f, es una característica única y especial indicada con la letra f (ff:ff:ff:ff:ff:ff) encargada de enviar la trama a todos los dispositivos de la red. A continuación, la dirección de origen se trata de VMware, una máquina virtual junto a su dirección Mac a diferencia del destino se tarta de unidifusión, es decir, de un único emisor a un único receptor.

El tipo de trama es determinado con 0x0806 este valor indica que es utilizado el protocolo ARP para la resolución de direcciones.

Para acabar con este apartado nos aparece el padding, un relleno que representa la cantidad de bits o bytes agregados para formar una cantidad par de bits o bytes que normalmente se establece en 0.

Trama ARP: Análisis Address Resolution Protocol[2]

```
Address Resolution Protocol (request)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: request (1)
    Sender MAC address: VMware_5e:1f:87 (00:0c:29:5e:1f:87)
    Sender IP address: 10.0.0.2
    Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
    Target IP address: 10.0.3.116
```

En este apartado, podemos ver información como la arquitectura anterior 0x0800 referente al protocolo ipv4, dirección Mac y dirección ip del dispositivo de destino (DNS) al igual que el origen (PC).

Los indicadores de hardware type y hardware size hacen referencia a la dirección de 6 bytes con la que son responde la trama del protocolo ARP que contiene 4bits. Al ser una trama sin respuesta, solo es una petición,

[1] https://www.youtube.com/watch?v=shp42M7gbDE
https://sites.google.com/site/herracolaboracion2/uso-de-wireshark-para-examinar-las-tramas-de-ethernet

[2] https://www.practicalnetworking.net/series/arp/traditional-arp/#:~:text=Hardware%20Type%20and%20Hardware%20Size,4%20bytes%20(32%20bits).
https://hazlolinux.com/wireshark/analisis-de-paquetes-arp-con-wireshark/

obtendremos el tipo 1 al igual que el opcode que en caso de ser respondido como la segunda trama analizada tendrá un 2.

Trama ARP 2: Diferencia

La segunda trama capturada obtendrá la misma información, pero con matices diferentes debido a ser la respuesta a la petición como la dirección origen que será el propio Gateway (DNS se encuentra en la misma maquina) y la dirección de destino es nuestra máquina virtual. Otra diferencia es el ya comentado opcode dentro del ARP además de al ser diferente el origen-destino las direcciones que obtendremos tanto Mac como de ip serán las correspondidas.

```
Sender MAC address: CyberTAN_72:3e:59 (00:45:e2:72:3e:59)
Sender IP address: 10.0.3.116
Target MAC address: VMware_5e:1f:87 (00:0c:29:5e:1f:87)
Target IP address: 10.0.0.2
```

Trama DNS: Análisis Ethernet

```
283 5.222862   10.0.3.116   10.0.0.2     DNS    75 Standard query 0x886f A www.mytcpip.com
340 5.551550   10.0.0.2     10.0.3.116   DNS   105 Standard query response 0x886f A www.mytcpip.com CNAME mytcpip.com A 35.214.245.118
```

De nuevo, observándolo de forma global podemos ver dos tramas DNS en las que preguntamos por la web en la inicial y nos devuelve una respuesta estándar con la dirección ip de la misma web en la segunda.

Centrándonos en la inicial podemos distinguir a la dirección de destino como el DNS y de dirección origen a nuestra maquina junto a sus respectivas Mac. Y a diferencia de las tramas anteriores esta tiene como tipo de protocolo ipv4 (0x0800).

```
Ethernet II, Src: CyberTAN_72:3e:59 (00:45:e2:72:3e
  > Destination: VMware_5e:1f:87 (00:0c:29:5e:1f:87)
  > Source: CyberTAN_72:3e:59 (00:45:e2:72:3e:59)
    Type: IPv4 (0x0800)
```

Trama DNS: Análisis Internet

```
v Internet Protocol Version 4, Src: 10.0.3.116, Dst: 10.0.0.2
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 61
    Identification: 0x49df (18911)
  > 000. .... = Flags: 0x0
    ...0 0000 0000 0000 = Fragment Offset: 0
    Time to Live: 128
    Protocol: UDP (17)
    Header Checksum: 0xd95b [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 10.0.3.116
    Destination Address: 10.0.0.2
```

Seguidamente, en la capa de red podemos ver que el protocolo destacado es el Ip en su versión 4 (ipv4) y principalmente nos muestra de nuevo las direcciones ip tanto de origen, como de destino. Además de filtros que muestran distintos tipos de servicios en el ámbito financiero (ECN[3]) y si tiene un code point (DSCP[4] es un segundo byte en la cabecera de los paquetes que diferencia la calidad de la comunicación sobre los datos que transportan), ambos casos son negativos por lo que observamos que aparecen con 0.

---

3 https://es.wikipedia.org/wiki/Electronic_communication_network#:~:text=El%20t%C3%A9rmino%20Electronic%20communication%20network,de%20las%20bolsas%20de%20valores.

4 https://es.wikipedia.org/wiki/Differentiated_Services_Code_Point

Continuando con información obtenida en esta capa también la cantidad de datos que transporta esta petición, 61, y tipos de flags pese a que no conlleva ninguno hay las siguientes posibilidades:

```
000. .... = Flags: 0x0
  0... .... = Reserved bit: Not set
  .0.. .... = Don't fragment: Not set
  ..0. .... = More fragments: Not set
...0 0000 0000 0000 = Fragment Offset: 0
```

Para terminar, es un dato muy interesante el TTL (Time to live) estos son los saltos que un paquete debe realizar para existir en una red antes de que sea descartado, es decir, si da más saltos a los mencionado, en este caso 128, este se descartara debido a ser protocolo UDP no tendrá un mecanismo un control de errores y se debe tener en cuenta que después del envío del paquete no recibe un confirmante por así decirlo.

Trama DNS: Análisis User Datagram Protocol[5]

UDP es un protocolo de red utilizado principalmente para establecer conexiones de baja latencia y tolerancia. Se utiliza principalmente en la comunicación sensible al tiempo como el DNS dentro de la capa de transporte.

```
User Datagram Protocol, Src Port: 51972, Dst Port: 53
    Source Port: 51972
    Destination Port: 53
    Length: 41
    Checksum: 0x55fb [unverified]
    [Checksum Status: Unverified]
    [Stream index: 74]
  > [Timestamps]
    UDP payload (33 bytes)
```

Puerto que utiliza este protocolo el es 53 y cómo podemos observar no consta de un checksum, únicamente los integrados en la IP.

El stream index se cuenta por la conversación basada en los sockets, puntos finales de la conexión UDP. Por lo tanto, cada par es un índice de transmisión. Teniendo en cuenta que los datos útiles transmitidos de playload son de 33bytes.[6]

Trama DNS: Análisis Domain Name System[7]

Finalmente, obtenemos el apartado para el protocolo DNS encargado de resolver nombres en la red para conocer la dirección ip mostrándonos una pregunta debido a que le hemos preguntado al DNS la dirección ip de www.mytcpip.com.

```
Domain Name System (query)
    Transaction ID: 0x886f
  > Flags: 0x0100 Standard query
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 0
  v Queries
    v www.mytcpip.com: type A, class IN
        Name: www.mytcpip.com
        [Name Length: 15]
        [Label Count: 3]
        Type: A (Host Address) (1)
        Class: IN (0x0001)
    [Response In: 340]
```

Se expone datos como el número de caracteres de la dirección como el tipo A debido a que hace referencia a la dirección ipv4 de un servidor web[8]. Clase IN es la clase del registra establecido en internet para registros como este que involucran un nombre de host.

RRs hace referencia los registros de recursos y es la unidad de entrada de información en los archivos de zona DNS.[9]

[5] https://vasexperts.com/es/resources/glossary/udp/
[6] https://ayuda.acens.com/hc/es/articles/360018220377--Qu%C3%A9-es-Payload-
[7] https://vasexperts.com/es/resources/glossary/udp/
[8] https://www.redeszone.net/tutoriales/internet/que-es-registros-dns/
[9] https://learn.microsoft.com/es-es/windows/win32/dns/resource-records

Trama DNS 2: Diferencia

La principal diferencia a la trama anterior es que esta se encuentra con el origen y destino inversos.

Source	Destination
10.0.0.2	10.0.3.116

Debido a que se trata de la respuesta del DNS a nuestra maquina ofreciéndole la dirección ip de la web podemos ver pequeños detalles en la capa de red como los datos de esta respuesta que superan a la anterior `Total Length: 91` .

En cuanto al UDP también parte que el puerto de origen es el suyo mismo y el de destino es nuestra maquina causado porque nos llega a nosotros la petición.

```
User Datagram Protocol, Src Port: 53, Dst Port: 51972
   Source Port: 53
   Destination Port: 51972
   Length: 71
   Checksum: 0xaa78 [unverified]
   [Checksum Status: Unverified]
   [Stream index: 74]
 ⌄ [Timestamps]
      [Time since first frame: 0.328688000 seconds]
      [Time since previous frame: 0.328688000 seconds]
   UDP payload (63 bytes)
```

Como podemos comprobar en la imagen ha aumentado respecto a la otra trama los datos de longitud, playload y los sockets del UDP que constituyen el stream index.

Los timestamps que aparecían en o anteriormente han aumentado gracias a que se han transportado la información en segundos de la captura.

Para terminar, en el apartado de DNS podremos ver que se hace una pregunta y esta obtiene dos respuestas de RRs es decir queda registrado su nombre e ip.

```
341 5.554970    10.0.3.116        35.214.245.118    ICMP    342 Echo (ping) request  id=0x0001, seq=4/1024, ttl=80 (no response found!)
342 5.612885    35.214.245.118    10.0.3.116        ICMP    110 Echo (ping) reply    id=0x0001, seq=4/1024, ttl=52
401 6.735071    10.0.3.116        35.214.245.118    ICMP    342 Echo (ping) request  id=0x0001, seq=5/1280, ttl=80 (no response found!)
424 6.790213    35.214.245.118    10.0.3.116        ICMP    110 Echo (ping) reply    id=0x0001, seq=5/1280, ttl=52
496 7.865486    10.0.3.116        35.214.245.118    ICMP    342 Echo (ping) request  id=0x0001, seq=6/1536, ttl=80 (no response found!)
498 7.928882    35.214.245.118    10.0.3.116        ICMP    110 Echo (ping) reply    id=0x0001, seq=6/1536, ttl=52
```

Por último, por protocolo ICMP veremos que realizaremos 3 envíos de paquetes tal y como maraca la orden. Con el *No response found* se refiere a que el mensaje como tal no ha sido respondido, pero este aun así ha sido recibido y respondido.

Trama ICMP Request: Análisis Ethernet

Para las tramas de solicitud nos encontraremos que la dirección origen será nuestro pc intentando hacer la conexión hacia la web mediante ipv4.

```
Ethernet II, Src: CyberTAN_72:3e:59 (00:45:e2:72:3e:5
 › Destination: Ubiquiti_15:27:81 (f0:9f:c2:15:27:81)
 › Source: CyberTAN_72:3e:59 (00:45:e2:72:3e:59)
   Type: IPv4 (0x0800)
```

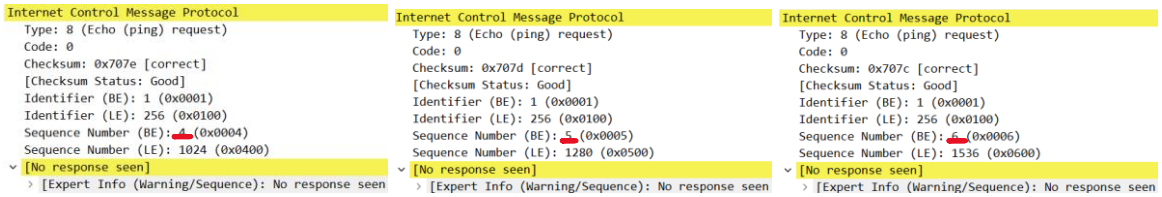
Trama ICMP: Análisis Internet

En la capa de internet nos encontramos de nuevo con el protocolo de ip en su versión 4 sin filtros ECN ni DSCP por lo que observamos que aparecen con 0. Al ser una petición de conexión en vez de un segmento de datos simple ya vemos que la longitud en bits es mayor y tiene un límite más restrictivo de TTL, 80.

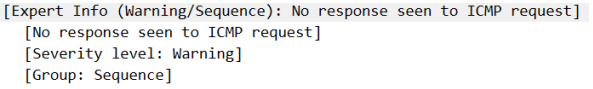
```
Internet Protocol Version 4, Src: 10.0.3.116, Ds
   0100 .... = Version: 4
   .... 0101 = Header Length: 20 bytes (5)
 › Differentiated Services Field: 0x00 (DSCP: CS
   Total Length: 328
   Identification: 0xb8b7 (47287)
 › 000. .... = Flags: 0x0
   ...0 0000 0000 0000 = Fragment Offset: 0
   Time to Live: 80
   Protocol: ICMP (1)
   Header Checksum: 0x8a3d [validation disabled]
   [Header checksum status: Unverified]
   Source Address: 10.0.3.116
   Destination Address: 35.214.245.118
```

Trama ICMP: Análisis Internet Control Message Protocol

En esta capa es donde se realiza la conexión como tal haciendo un ping y en checksum observamos que se realiza de forma exitosa. Todos estos paquetes de datos son enviados con un identificador llamado número de secuencia para que al llegar a su destino sepan cual es el orden por lo que en la primera trama de ping veremos el identificador 4 tanto en el request como reply porque hacen referencia al mismo, pero para el siguiente este número será 5 y al tercer paquete encontraremos un 6 siempre con la misma longitud de información request=68 bytes y reply=300 bytes.



Aunque estos no hayan sido respondidos podemos ver los apartados de información más experta en la que se menciona si ha habido respuesta, si pertenece a uno individual o a una sequencia y el nivel de seguridad.

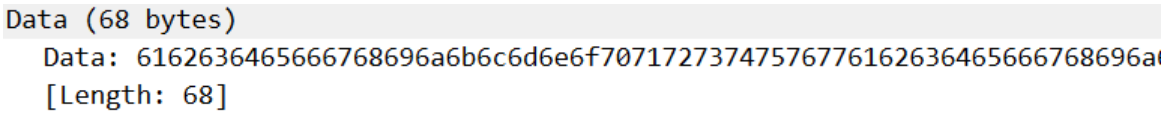


Trama ICMP Reply: Diferencia

Para las tramas de respuesta en el apartado de Macs veremos que son inversos a la petición debido a que nos está respondiendo la web a nosotros por lo que nuestro pc será la dirección de destino.

En el apartado de IP veremos que el encabezado de la respuesta contiene 20bytes y el TTL de este paquete será todavía más limitado al anterior siendo 52.

Para terminar el análisis, en el apartado de ICMP veremos que el tipo o hace referencia a la respuesta con un checksum correcto indicando que fue enviado y recibido de forma exitosa. La diferencia más notoria es el apartado de datos dentro de este protocolo que contiene una longitud de 68b con la respuesta al intento de conexión. Hay que recalcar que, aunque el mensaje en si mismo no fuera respuesta en su esencia si se establece la conexión al ver que el mensaje le ha sido enviado y respondido.



PRACTICA CMD:

Para el script de CMD se realizarán las mismas tramas para la conexión que en el sscript de powershell pero hay pequeños matices que la hacen diferente además de otras curiosidades que se han mostrado.

En primer lugar, ejecutamos el script en una interfaz de red móvil en la que no hay otros dispositivos realizando peticiones y podemos observar cómo constantemente el Gateway

va preguntado a todas las direcciones ip para tenerlas anotadas en su caché por lo que nos encontraremos una gran variedad de tramas preguntando por nuestra dirección para que la tenga la puerta de enlace.

235 4.016280	CyberTAN_72:3e:59	Broadcast	ARP	42 Who has 10.0.0.2? Tell 10.0.3.116
236 4.036532	VMware_5e:1f:87	CyberTAN_72:3e:59	ARP	60 10.0.0.2 is at 00:0c:29:5e:1f:87

La principal curiosidad[10] es en el apartado de DNS donde en vez de únicamente encontrarse la trama de petición para tener la dirección ip de la web nos encontramos además gran multitud de tramas debido a la opción de Resolve Destination en la que el cmdlet intenta resolver el nombre DNS del destino cuando se usa junto al parámetro de traceroute haciendo así que todos los hosts intermedios también se recuperarán. Esto quiere decir que todos los aplicativos y navegaciones realizadas en su momento, aunque estén en segundo plano se anotaran.

3378 232.649982	192.168.43.48	192.168.43.1	DNS	91	Standard query 0x13ef A teams.events.data.microsoft.com
3379 232.744839	192.168.43.48	192.168.43.1	DNS	91	Standard query 0x13ef A teams.events.data.microsoft.com
3380 232.947195	192.168.43.1	192.168.43.48	DNS	210	Standard query response 0x13ef A teams.events.data.microsoft.com CNAME teams-events-data.trafficma

[11]

Sin entrar mucho en detalle podremos ver con el filtro de TCP que los aplicativos como el anterior se encontraran aquí por su navegación.

TLSv1.2	1454	Application Data, Application Data		
TCP	1454	443 → 50496 [ACK] Seq=58921 Ack=1353 Win=4193024 Len=1400 [TCP segment of a reassembled PDU]		
TCP	54	50496 → 443 [ACK] Seq=1353 Ack=60321 Win=131584 Len=0		

Trama TCP: Análisis Ethernet

Para terminar, en este script enviamos 1 segmento de paquetes de 2 tramas:

220 3.790668	13.107.6.171	10.0.3.116	TCP	60 443 → 49804 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
345 5.542413	13.107.136.9	10.0.3.116	TCP	60 443 → 49798 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0

```
Ethernet II, Src: Ubiquiti_15:27:81 (f0:9f:c2:15:27:81)
  > Destination: CyberTAN_72:3e:59 (00:45:e2:72:3e:59)
  > Source: Ubiquiti_15:27:81 (f0:9f:c2:15:27:81)
    Type: IPv4 (0x0800)
    Padding: 000000000000
```

De forma global podemos comprobar que se ha enviado 1 trama para cada paquete en la que nuestro pc es el receptor a través del protocolo TCP utilizando ipv4 y se incluye la combinación de dos paquetes tanto la recepción del mensaje (ACK: reconocimiento/ confirmación de recepción) como la notificación de que se ha cerrado la conexión con el RST.[12]

Trama TCP: Análisis Internet

```
Internet Protocol Version 4, Src: 13.107.6.171, Dst: 10.0.3.116
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
> Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
  Total Length: 40
  Identification: 0xd1a2 (53666)
> 010. .... = Flags: 0x2, Don't fragment
  ...0 0000 0000 0000 = Fragment Offset: 0
  Time to Live: 109
  Protocol: TCP (6)
  Header Checksum: 0x1aa4 [validation disabled]
  [Header checksum status: Unverified]
  Source Address: 13.107.6.171
  Destination Address: 10.0.3.116
```

En el apartado de IP veremos que ultilizamos la versión 4 con una cabecera de 20bytes sin fragmentar

```
> 010. .... = Flags: 0x2, Don't fragment
    0... .... = Reserved bit: Not set
    .1.. .... = Don't fragment: Set
    ..0. .... = More fragments: Not set
  ...0 0000 0000 0000 = Fragment Offset: 0
```

Y con un TTL de 109 saltos antes de la extinción de este paquete.

[10] https://learn.microsoft.com/es-es/powershell/module/microsoft.powershell.management/test-connection?view=powershell-7.2

[11] Ejemplo del Microsoft teams abierto en segundo plano.

[12] https://respuestasrapidas.com.mx/que-es-un-rst-ack/#Que_es_un_RST_ACK

Trama TCP: Análisis Transmission Control Protocol

Como se puede comprobar en la imagen el paquete es enviado a través de puerto 443 utilizado para la navegación web además del protocolo HTTPS que es seguro y utiliza el TLS por debajo.

Se indica que la conversación es incompleta porque no ha respondido como tal la web a la que intentamos la conexión, pero eso no indica que sea incorrecta o que no se alcance.

De la misma manera que enviamos un paquete antes que el otro se comprara cual ha sido el orden con el stream index. Al ser el mismo tipo de paquete tendrán la misma longitud.

```
Transmission Control Protocol, Src Port: 443, Dst Port: 49804
   Source Port: 443
   Destination Port: 49804
   [Stream index: 1]
   [Conversation completeness: Incomplete (36)]
   [TCP Segment Len: 0]
   Sequence Number: 1    (relative sequence number)
   Sequence Number (raw): 4146454299
   [Next Sequence Number: 1    (relative sequence number)]
   Acknowledgment Number: 1    (relative ack number)
   Acknowledgment number (raw): 3918459188
   0101 .... = Header Length: 20 bytes (5)
```

```
Transmission Control Protocol, Src Port: 443, Dst Port: 4979
   Source Port: 443
   Destination Port: 49798
   [Stream index: 2]
   [Conversation completeness: Incomplete (36)]
   [TCP Segment Len: 0]
   Sequence Number: 1    (relative sequence number)
   Sequence Number (raw): 3622804752
   [Next Sequence Number: 1    (relative sequence number)]
   Acknowledgment Number: 1    (relative ack number)
   Acknowledgment number (raw): 2143585401
   0101 .... = Header Length: 20 bytes (5)
```

Aunque en esta ocasión no se nos proporcione mucha información valida podemos ver varios opciones de confirmación como el checksum, la medida que ha ocupado la ventana, en este caso no ha habido debido a que ha sido en segundo plano, de ahí el -1 y las marcas de tiempo.

```
Flags: 0x014 (RST, ACK)
Window: 0
[Calculated window size: 0]
[Window size scaling factor: -1 (unknown)]
Checksum: 0xcf68 [unverified]
[Checksum Status: Unverified]
Urgent Pointer: 0
[Timestamps]
   [Time since first frame in this TCP stream: 0.000000000 seconds]
   [Time since previous frame in this TCP stream: 0.000000000 seconds]
```