

Javier Alejandro Prado Ramírez - 21486

## 1.1 Personalización del entorno

En la primera parte se realizará la personalización del entorno de Wireshark, de modo que se adapte a nuestras preferencias de uso.

1. Inicie Wireshark

2. Cree un perfil con su primer nombre y primer apellido (edit -> configuration profile)

Perfil	Tipo	Auto Switch Filter
Default	Predeterminado	
Bluetooth	Global	—
Classic	Global	—
No Reassembly	Global	—
Javier Prado	Personal	

3. Descargue el archivo <https://www.cloudshark.org/captures/e6fb36096dbb> (Export -> Download)

4. Abra el archivo descargado, el archivo contiene transmisiones capturadas, y existen diversas columnas que representan la data.

5. Aplique el formato de tiempo Time of Day (view -> Time Display Format)

6. Agregue una columna con la longitud del protocolo (preferences -> column -> +)

7. Elimine u oculte la columna Longitud (click derecho -> desmarcar columna)

No.	Time	Source	Destination	Protocol	TCP Segment Len	Info
1	11:16:47.126585	192.168.0.46	172.67.75.39	TCP	0	51111 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
2	11:16:47.177831	172.67.75.39	192.168.0.46	TCP	0	443 → 51111 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1400 SACK_PERM WS
3	11:16:47.177959	192.168.0.46	172.67.75.39	TCP	0	51111 → 443 [ACK] Seq=1 Ack=1 Win=131584 Len=0
4	11:16:47.178243	192.168.0.46	172.67.75.39	TLSv1.3	517	Client Hello (SNI=www.wireshark.org)
5	11:16:47.255959	172.67.75.39	192.168.0.46	TCP	0	443 → 51111 [ACK] Seq=1 Ack=518 Win=67584 Len=0
6	11:16:47.267905	172.67.75.39	192.168.0.46	TLSv1.3	1460	Server Hello, Change Cipher Spec
7	11:16:47.267905	172.67.75.39	192.168.0.46	TLSv1.3	348	Application Data
8	11:16:47.267999	192.168.0.46	172.67.75.39	TCP	0	51111 → 443 [ACK] Seq=518 Ack=1809 Win=131584 Len=0
9	11:16:49.048600	192.168.0.46	172.67.75.39	TLSv1.3	64	Change Cipher Spec, Application Data
10	11:16:49.048918	192.168.0.46	172.67.75.39	TLSv1.3	92	Application Data
11	11:16:49.049309	192.168.0.46	172.67.75.39	TLSv1.3	666	Application Data
12	11:16:49.073867	172.67.75.39	192.168.0.46	TCP	0	443 → 51111 [ACK] Seq=1809 Ack=582 Win=67584 Len=0
13	11:16:49.073867	172.67.75.39	192.168.0.46	TLSv1.3	528	Application Data, Application Data
14	11:16:49.074157	192.168.0.46	172.67.75.39	TLSv1.3	31	Application Data
15	11:16:49.079528	172.67.75.39	192.168.0.46	TCP	0	443 → 51111 [ACK] Seq=2337 Ack=674 Win=67584 Len=0
16	11:16:49.079528	172.67.75.39	192.168.0.46	TCP	0	443 → 51111 [ACK] Seq=2337 Ack=1340 Win=68608 Len=0
17	11:16:49.079528	172.67.75.39	192.168.0.46	TLSv1.3	31	Application Data

8. Aplique un esquema de paneles que sea de su preferencia (que no sea el esquema por defecto) (preferences -> Layout)

Archivo Edición Visualización Ir Captura Analizar Estadísticas Telefonía Wireless Herramientas Ayuda

Aplique un filtro de visualización... «Ctrl-F»

No.	Time	Source	Destination	Protocol	TCP Segment Len	Info
9	11:16:49.048600	192.168.0.46	172.67.75.39	TLSv1.3	64	Change Cipher Spec, Application Data
10	11:16:49.048918	192.168.0.46	172.67.75.39	TLSv1.3	92	Application Data
11	11:16:49.049309	192.168.0.46	172.67.75.39	TLSv1.3	666	Application Data
12	11:16:49.073867	172.67.75.39	192.168.0.46	TCP	0	443 → 51111 [ACK] Seq=1809 Ack=582 Win=67584 Len=0
13	11:16:49.073867	172.67.75.39	192.168.0.46	TLSv1.3	528	Application Data, Application Data
14	11:16:49.074157	192.168.0.46	172.67.75.39	TLSv1.3	31	Application Data
15	11:16:49.079528	172.67.75.39	192.168.0.46	TCP	0	443 → 51111 [ACK] Seq=2337 Ack=674 Win=67584 Len=0
16	11:16:49.079528	172.67.75.39	192.168.0.46	TCP	0	443 → 51111 [ACK] Seq=2337 Ack=1340 Win=68608 Len=0
17	11:16:49.079528	172.67.75.39	192.168.0.46	TLSv1.3	31	Application Data
18	11:16:49.099770	172.67.75.39	192.168.0.46	TCP	0	443 → 51111 [ACK] Seq=2368 Ack=1371 Win=68608 Len=0

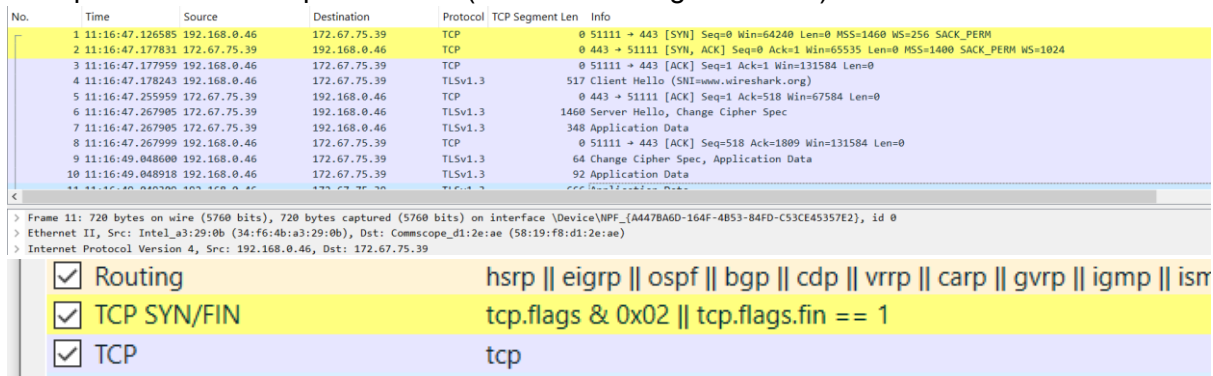
> Frame 12: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface \Device\NPF\_{A447B6D-164F-4B53-84FD-C53CE45357E2}, id 0  
Ethernet II, Src: Comscope\_d12e:ae (58:19:f8:d1:2e:ae), Dst: Intel\_a3:29:0b (34:f6:4b:a3:29:0b)  
> Internet Protocol Version 4, Src: 172.67.75.39, Dst: 192.168.0.46  
Transmission Control Protocol, Src Port: 443, Dst Port: 51111, Seq: 1809, Ack: 582, Len: 0  
Source Port: 443  
Destination Port: 51111  
[Stream index: 0]  
[Stream Packet Number: 12]  
> [Conversation completeness: Incomplete, DATA (15)]  
[TCP Segment Len: 0]  
Sequence Number: 1809 (relative sequence number)  
Sequence Number (raw): 3298758551  
[Next Sequence Number: 1809 (relative sequence number)]

```

0000 34 f6 4b a3 29 0b 58 19 f8 d1 2e ae 08 00 45 00 4 K-X:.....E-
0010 00 28 a5 52 40 00 3b 06 e2 3c ac 43 4b 27 c0 a8  (R):<CKT...
0020 00 2e 01 b6 c7 a7 c4 9f 0f 97 0e a7 8c 73 50 10 .....G:SP-
0030 00 42 be fd 00 00 00 00 00 00 00 00 00 00 00 00 B:.....

```

9. Aplique una regla de color para el protocolo TCP cuyas banderas SYN sean iguales a 1, y coloque el color de su preferencia. (View -> coloring rules -> +)



The image shows the Wireshark packet list with several TCP packets. Below the list, the 'Coloring Rules' dialog is open, showing a rule for 'tcp.flags.syn == 1' with a yellow background color selected.

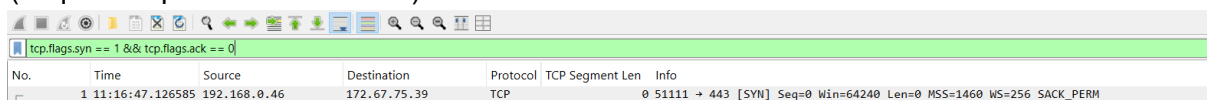
No.	Time	Source	Destination	Protocol	TCP Segment Len	Info
1	11:16:47.126585	192.168.0.46	172.67.75.39	TCP	0	51111 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
2	11:16:47.177831	172.67.75.39	192.168.0.46	TCP	0	443 → 51111 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1400 SACK_PERM WS=1024
3	11:16:47.177959	192.168.0.46	172.67.75.39	TCP	0	51111 → 443 [ACK] Seq=1 Ack=1 Win=131584 Len=0
4	11:16:47.178243	192.168.0.46	172.67.75.39	TLSv1.3		517 Client Hello (SHA256WithRSA) (org)
5	11:16:47.255959	172.67.75.39	192.168.0.46	TCP	0	443 → 51111 [ACK] Seq=1 Ack=518 Win=67584 Len=0
6	11:16:47.267905	172.67.75.39	192.168.0.46	TLSv1.3		1460 Server Hello, Change Cipher Spec
7	11:16:47.267905	172.67.75.39	192.168.0.46	TLSv1.3		348 Application Data
8	11:16:47.267999	192.168.0.46	172.67.75.39	TCP	0	51111 → 443 [ACK] Seq=518 Ack=1809 Win=131584 Len=0
9	11:16:49.048600	192.168.0.46	172.67.75.39	TLSv1.3		64 Change Cipher Spec, Application Data
10	11:16:49.048918	192.168.0.46	172.67.75.39	TLSv1.3		92 Application Data

Coloring Rules:

- ☒ Routing
- ☒ TCP SYN/FIN
- ☒ TCP

Rule: tcp.flags.syn == 1 && tcp.flags.ack == 0

10. Cree un botón que aplique un filtro para paquetes TCP con la bandera SYN igual a 1. (esquina superior derecha -> +)

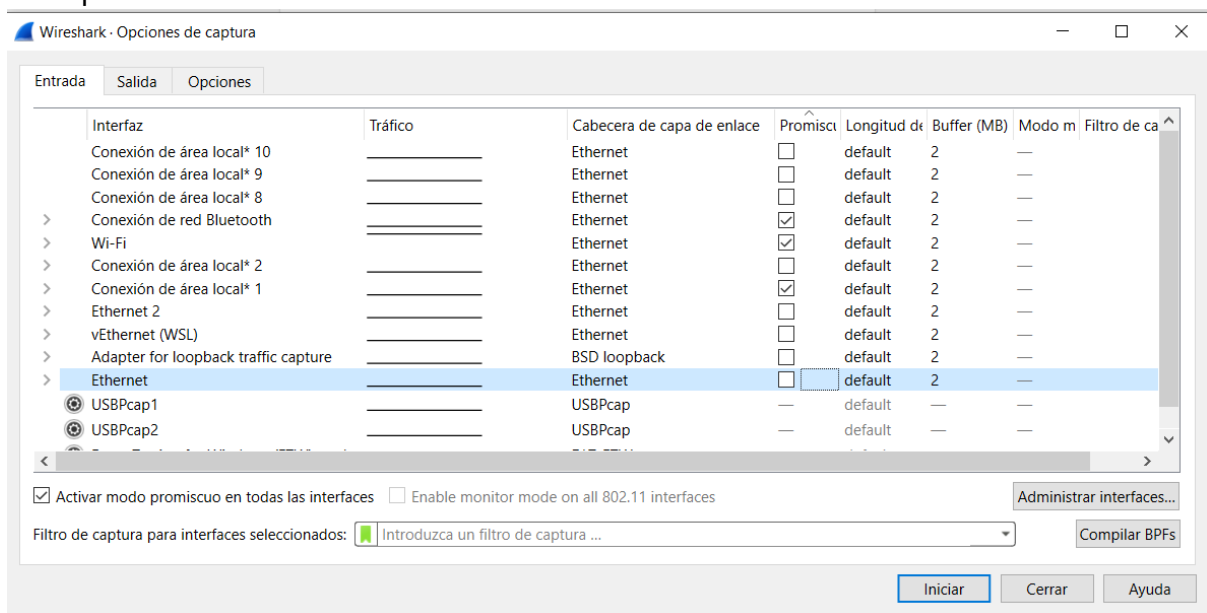


The image shows the Wireshark packet list with a filter applied: 'tcp.flags.syn == 1 && tcp.flags.ack == 0'. The filter is highlighted in green.

No.	Time	Source	Destination	Protocol	TCP Segment Len	Info
1	11:16:47.126585	192.168.0.46	172.67.75.39	TCP	0	51111 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM

11. Oculte las interfaces virtuales (en caso aplique: capture -> options)

Se debe realizar tomas de pantalla que muestren el entorno final personalizado, el nombre del perfil y el uso de las regla de color y botón del filtro, así como la lista simplificada de las interfaces de captura.



The image shows the 'Wireshark - Opciones de captura' dialog box. The 'Entrada' tab is selected, showing a list of interfaces. The 'Ethernet' interface is selected, and the 'Promiscuo' checkbox is checked. The 'Filtro de captura' field is empty.

Entrada	Salida	Opciones
Interfaz	Tráfico	Cabecera de capa de enlace
Conexión de área local* 10		Ethernet
Conexión de área local* 9		Ethernet
Conexión de área local* 8		Ethernet
> Conexión de red Bluetooth		Ethernet
> Wi-Fi		Ethernet
> Conexión de área local* 2		Ethernet
> Conexión de área local* 1		Ethernet
> Ethernet 2		Ethernet
> vEthernet (WSL)		Ethernet
> Adapter for loopback traffic capture		BSD loopback
> Ethernet		Ethernet
USBPCap1		USBPCap
USBPCap2		USBPCap

Activar modo promiscuo en todas las interfaces ☒ Enable monitor mode on all 802.11 interfaces ☐

Filtro de captura para interfaces seleccionados:

Botones: Iniciar, Cerrar, Ayuda

## 1.2 Configuración de la captura de paquetes

En la segunda parte, se realizará una captura de paquetes con un ring buffer.

1. Abra una terminal y ejecute el comando `ifconfig/ipconfig` (dependiendo de su OS).

Detalle y explique lo observado, investigue (i.e.: 'man ifconfig', documentación) de ser necesario.

```
Sufijo DNS específico para la conexión. . . :
Vínculo: dirección IPv6 local. . . : fe80::424
Dirección IPv4. . . . . : 19
Máscara de subred . . . . . : 25
Puerta de enlace predeterminada . . . . . :
```

2. Luego, retornando a Wireshark, desactive las interfaces virtuales o que no aplique.

Interfaz	Tráfico	Cabecera de capa de enlace	Promisc	Longitud de	Buffer (MB)	Mod
WAN Miniport (Network Monitor): Conexión de área local* 10	—	Ethernet	<input type="checkbox"/>	default	2	—
WAN Miniport (IPv6): Conexión de área local* 9	—	Ethernet	<input type="checkbox"/>	default	2	—
WAN Miniport (IP): Conexión de área local* 8	—	Ethernet	<input type="checkbox"/>	default	2	—
> Intel(R) Wi-Fi 6 AX201 160MHz: Wi-Fi	↕	Ethernet	<input checked="" type="checkbox"/>	default	2	—

3. Realice una captura de paquetes con la interfaz de Ethernet o WiFi con una configuración de ring buffer, con un tamaño de 5 MB por archivo y un número máximo de 10 archivos (puede hacerlo por medio de la interfaz de usuario o por medio de comandos) Genere tráfico para que los archivos se creen. Defina el nombre de los archivos de la siguiente forma: lab1\_carnet.pgcap (options -> capture -> output) Se debe realizar tomas de pantalla de la configuración o comandos para la creación del ring buffer, así como los archivos generados.

Entrada
Salida
Opciones

Capturar a archivo permanente

Archivo: C:\Users\HP\Documents\U\Redes\lab1\_21486
Explorar...

Formato de salida: ☒ pcapng ☐ pcap

☐ Crear un nuevo archivo automáticamente...

☐ después de 100000 paquetes

☐ después de 1 kilobytes

☐ después de 1 segundos

☐ cuando el tiempo es múltiplo de 1 horas

compresión: ☒ Ninguna ☐ gzip

File infix pattern: ☒ YYYYmmDDHHMMSS\_NNNNN ☐ NNNNN\_YYYYmmDDHHMMSS

☒ Usar un buffer cíclico con 2 archivos

Iniciar Cerrar Ayuda

```

À M<+  ÿÿÿÿÿÿ 7 Intel(R) Core(TM) i7-10750H CPU @ 2.60GHz (with SSE4.2)  % 64-bit Windows 10 (22H2
d ïÿÿÿÿÿ  ðM-SEARCH * HTTP/1.1
Host: 239.255.255.250:1900
ST: urn:schemas-upnp-org:device:InternetGatewayDevice:1
Man: "ssdp:discover"
MX: 3

```

```

õ  ò  è9  ÉMÿÖ  ò  ÌÜ-Ehíðó%kv E^  Áá>  Úý
d ïÿÿÿÿÿ  ðÉM-SEARCH * HTTP/1.1
HOST: 239.255.255.250:1900
MAN: "ssdp:discover"
MX: 1
ST: urn:dial-multiscreen-org:service:dial:1
USER-AGENT: Chromium/134.0.6998.178 Windows

```

```

õ  Ì  Ì  è9  íÄÿÖ  ©  ÌÜ-Ehíðf+Tm E^  >Ä-  ÉÁ
d ïÿÿÿÿÿ  ðÉM-SEARCH * HTTP/1.1
Host: 239.255.255.250:1900
ST: uuid:954de7a6-7395-44e9-8ff1-c4dc7df9bf1f
Man: "ssdp:discover"
MX: 3

```

### 1.3 Análisis de paquetes

En la tercera parte se analizará el protocolo HTTP. Debe realizar tomas de pantalla que validen sus respuestas.

1. Abra su navegador, inicie una captura de paquetes en Wireshark (sin filtro) en la interfaz y acceda a la siguiente dirección: <http://gaia.cs.umass.edu/wireshark-labs/INTRO-wireshark-file1.html>
2. Detenga la captura de paquetes (si desea realizar una nueva captura de la página deberá borrar el caché de su navegador, de lo contrario no se realizará la captura del protocolo HTTP).

http2	17:47.259041	10.100.1.60	128.119.245.12	HTTP	533 GET /wireshark-labs/INTRO-wireshark-file1.html HTTP/1.1
http3	17:47.362926	128.119.245.12	10.100.1.60	HTTP	438 HTTP/1.1 200 OK (text/html)
289	17:17:47.435559	10.100.1.60	128.119.245.12	HTTP	479 GET /favicon.ico HTTP/1.1
302	17:17:47.560968	128.119.245.12	10.100.1.60	HTTP	484 HTTP/1.1 404 Not Found (text/html)

3. Responda las siguientes preguntas:

a. ¿Qué versión de HTTP está ejecutando su navegador?

Http 1.1

b. ¿Qué versión de HTTP está ejecutando el servidor?

Http 1.1

c. ¿Qué lenguajes (si aplica) indica el navegador que acepta a el servidor?

Accept-encoding: gzip, deflate, br  
Accept-Language: es-419,es;q=0.9,es-ES;q=0.8,en;q=0.7,en-GB;q=0.6,en-US;q=0.5\r\n

d. ¿Cuántos bytes de contenido fueron devueltos por el servidor?

Se capturaron 533 bytes

e. En el caso que haya un problema de rendimiento mientras se descarga la página, ¿en que elementos de la red convendría “escuchar” los paquetes? ¿Es conveniente instalar Wireshark en el servidor? Justifique.

Es recomendable cuando nos e descarga algo de la web “escuchar” los paquetes para identificar el problema. Como escuchar el cliente para observar los tiempos de respuesta del servidor. Tambien el router para monitorear el tráfico que entra y sale de la red.

Conclusion

Wireshark es una herramienta poderosa y esencial para el análisis de redes, ya que permite capturar, inspeccionar y diagnosticar el tráfico que circula entre dispositivos en tiempo real. Su capacidad para descomponer paquetes en diferentes capas del modelo OSI lo convierte en una solución ideal para identificar fallos de rendimiento, problemas de seguridad o errores de configuración en la red