



Universidade Federal do Ceará

Disciplina: Redes de Computadores

Profa Atslands Rocha

Capturando o Way Handshake do TCP (Fonte: Kurose 6/7, colaboração parte 1: Profª Nídia Campos)

Objetivo:

Analizar as mensagens de estabelecimento de conexão do protocolo TCP.

Entregáveis: Cada aluno deve fazer o upload no Google ClassRoom das respostas às questões e o arquivo .pcap desta prática (extensão do packet tracer).

TAREFA 1 - Detectar o endereço IP do seu host

PASSO 1. Use o comando ipconfig para descobrir seu endereço IP da conexão local ou conexão sem fio.

A screenshot of a Windows Command Prompt window titled "Administrador: Prompt de Comando". The window shows the output of the "ipconfig" command. The output includes configuration details for several network adapters:

- Adaptador Ethernet Ethernet:
 - Estado da mídia: mídia desconectada
 - Sufixo DNS específico de conexão: .
- Adaptador de Rede sem Fio Conexão Local* 1:
 - Estado da mídia: mídia desconectada
 - Sufixo DNS específico de conexão: .
- Adaptador de Rede sem Fio Conexão Local* 3:
 - Estado da mídia: mídia desconectada
 - Sufixo DNS específico de conexão: .
- Adaptador de Rede sem Fio Wi-Fi:
 - Sufixo DNS específico de conexão: .
 - Endereço IPv6 de link local: fe80::30d0:36c1:7d59:46cc%22
 - Endereço IPv4: 192.168.0.104
 - Máscara de Sub-rede: 255.255.255.0
 - Gateway Padrão: 192.168.0.1

TAREFA 2 - Detectar o endereço IP do servidor de destino

PASSO 1. Escolha um site web para você abrir no navegador.

PASSO 2. Execute o comando nslookup para descobrir o endereço IP do servidor web. O servidor deve ter apenas um endereço IP.

```

C:\ Prompt de Comando
C:\Users\neand>nslookup gaia.cs.umass.edu
Servidor: UnKnown
Address: 192.168.0.1

Não é resposta autoritativa:
Nome: gaia.cs.umass.edu
Address: 128.119.245.12

C:\Users\neand>_

```

TAREFA 3 - Usar o navegador web para se estabelecer conexão TCP

PASSO 1 - Desabilite todas as interfaces de rede do seu host, exceto a interface de rede que possui o endereço IP registrado na Tarefa 1.

PASSO 2 - Configure o Wireshark para capturar mensagens.

PASSO 3 - No navegador web, abra o site escolhido.

PASSO 4 - Pare o Wireshark e salve o trace da captura em um arquivo .pcap.

TAREFA 4 - Analisar as três mensagens de estabelecimento de conexão do TCP

PASSO 1 - Na captura do Wireshark, filtre todas as mensagens que possui o IP do servidor web. Use o filtro ip.addr== ENDEREÇO_IP_DO_SERVIDOR_WEB

PASSO 2 - Localize a mensagem HTTP Request solicitando o site que você escolheu. As mensagens de apresentação do TCP (*Three-Way Handshake*) se encontram antes desse HTTP Request.

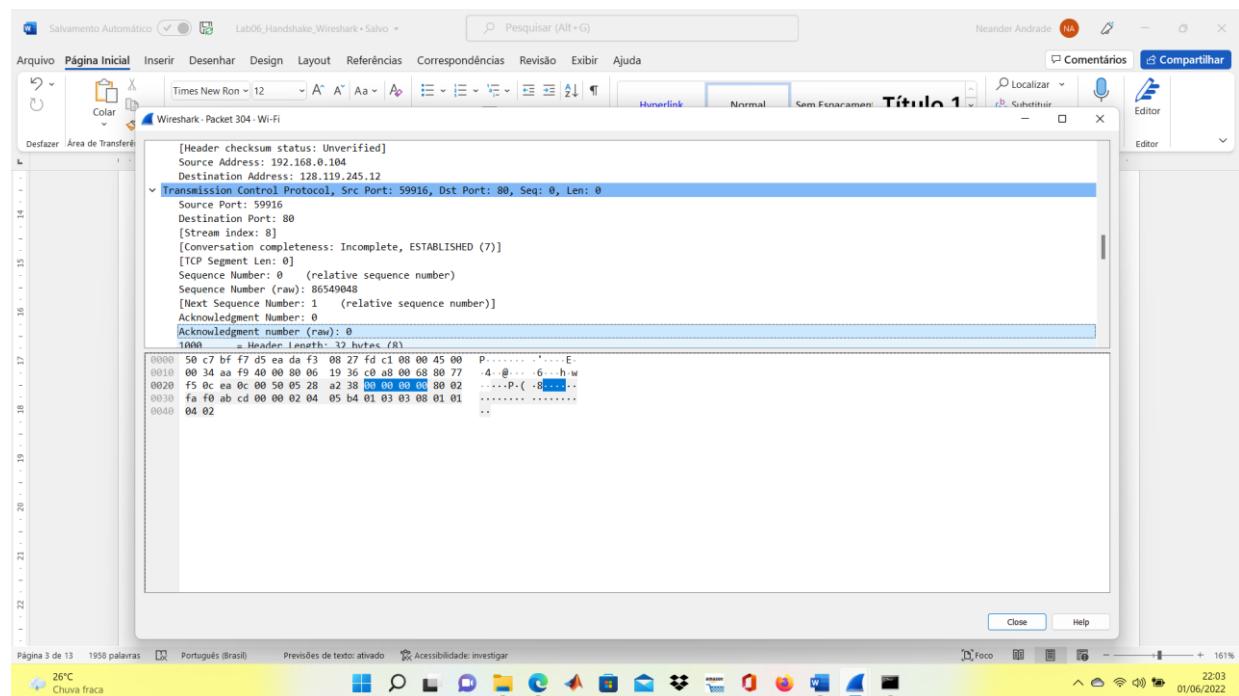
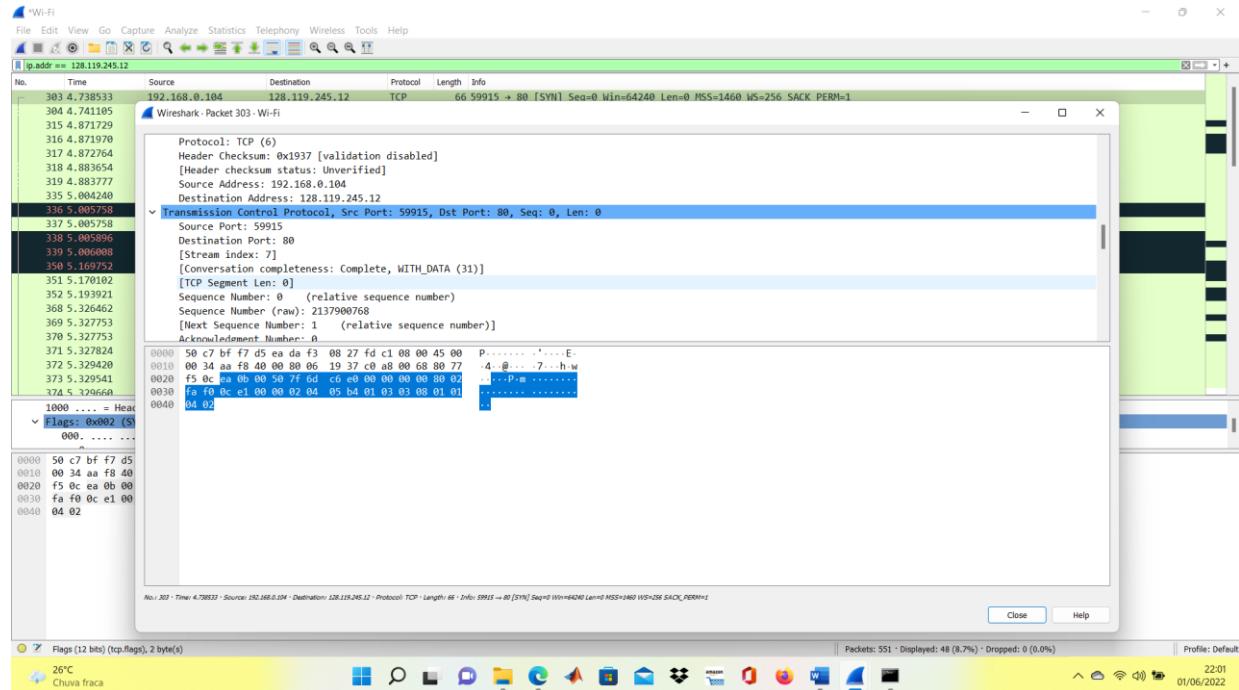
317 4.872764 192.168.0.104 128.119.245.12 HTTP 530 GET / HTTP/1.1

No.	Time	Source	Destination	Protocol	Length	Info
303	4.738533	192.168.0.104	128.119.245.12	TCP	66	59915 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
304	4.741105	192.168.0.104	128.119.245.12	TCP	66	59916 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
315	4.871729	128.119.245.12	192.168.0.104	TCP	66	80 → 59915 [SYN, ACK] Seq=0 Ack=1 Win=29208 Len=0 MSS=1412 SACK_PERM=1 WS=128
316	4.871970	192.168.0.104	128.119.245.12	TCP	54	59915 → 80 [ACK] Seq=1 Ack=1 Win=131072 Len=0
317	4.872764	192.168.0.104	128.119.245.12	HTTP	530	[GET / HTTP/1.1]
318	4.883654	128.119.245.12	192.168.0.104	TCP	66	80 → 59916 [SYN, ACK] Seq=0 Ack=1 Win=29208 Len=0 MSS=1412 SACK_PERM=1 WS=128
319	4.883777	192.168.0.104	128.119.245.12	TCP	54	59916 → 80 [ACK] Seq=1 Ack=1 Win=131072 Len=0
335	5.004240	128.119.245.12	192.168.0.104	TCP	54	80 → 59915 [ACK] Seq=1 Ack=1 Win=131072 Len=0
336	5.005758	128.119.245.12	192.168.0.104	HTTP	1466	[TCP Previous segment not captured] Continuation
337	5.005758	128.119.245.12	192.168.0.104	HTTP	241	Continuation
338	5.005896	192.168.0.104	128.119.245.12	TCP	66	[TCP Dup ACK 310#1] 59915 → 80 [ACK] Seq=477 Ack=1 Win=131072 Len=0 SLE=1413 SRE=2825
339	5.006008	192.168.0.104	128.119.245.12	TCP	66	[TCP Dup ACK 310#2] 59915 → 80 [ACK] Seq=477 Ack=1 Win=131072 Len=0 SLE=1413 SRE=3012
350	5.169752	128.119.245.12	192.168.0.104	TCP	1466	[TCP Retransmission] 80 → 59915 [ACK] Seq=2 Ack=477 Win=30336 Len=1412
351	5.170102	192.168.0.104	128.119.245.12	TCP	54	59915 → 80 [ACK] Seq=477 Ack=3012 Win=131072 Len=0
352	5.193921	192.168.0.104	128.119.245.12	HTTP	478	GET /corgi_imap.jpg HTTP/1.1
368	5.326462	128.119.245.12	192.168.0.104	TCP	1466	80 → 59915 [ACK] Seq=3012 Ack=901 Win=31360 Len=1412 [TCP segment of a reassembled PDU]
369	5.327753	128.119.245.12	192.168.0.104	TCP	1466	80 → 59915 [ACK] Seq=4424 Ack=901 Win=31360 Len=1412 [TCP segment of a reassembled PDU]
370	5.327753	128.119.245.12	192.168.0.104	TCP	1466	80 → 59915 [ACK] Seq=5836 Ack=901 Win=31360 Len=1412 [TCP segment of a reassembled PDU]
371	5.327824	192.168.0.104	128.119.245.12	TCP	54	59915 → 80 [ACK] Seq=901 Ack=7248 Win=131072 Len=0
372	5.329420	128.119.245.12	192.168.0.104	TCP	1466	80 → 59915 [ACK] Seq=7248 Ack=901 Win=31360 Len=1412 [TCP segment of a reassembled PDU]
373	5.329541	192.168.0.104	128.119.245.12	TCP	54	59915 → 80 [ACK] Seq=901 Ack=8660 Win=131072 Len=0
374	5.329660	128.119.245.12	192.168.0.104	TCP	1466	80 → 59915 [ACK] Seq=901 Ack=901 Win=131072 Len=1412 [TCP segment of a reassembled PDU]
8101	Header Length: 20 bytes (5)		
Flags: 0x0108 (PSH, ACK)					
0000	= Reserved: Not set		
0020	F5 00 ea 0b 00 50 7f 6d c6 e1 2f 8e 1b fd 50 18-P-m ..-/-..				
0030	02 00 8e c2 00 00 47 45 54 20 2f 20 84 54 54 50GE T / HTTP				
0040	2f 31 2e 31 0d 0a 48 6f 73 74 3a 20 67 61 69 63	/1.1-Ho st: gaia				
0050	2e 63 72 2e 75 6d 61 73 73 2e 65 64 75 0d 0a 43	.cs.umass.s.edu-C				
0060	6f 6e 65 63 7d 69 6f 3a 20 6b 65 65 70 2d	onnectio n: keep-				
0070	61 6c 69 76 65 0d 0a 55 70 67 72 61 64 65 2d 49	alive-U pgrade-I				
0080	6c 73 62 63 65 65 2d 52 65 71 64 65 2d 49	nnection Requests				
0090	6c 73 62 31 0a 55 70 67 65 66 74 3a	: 1.0.0-Agenc				
00a0	20 4d 6f 7a 69 6c 6c 61 2f 35 2e 30 20 28 57 69	Mozilla/5.0 (Wi				
00b0	6e 64 6f 77 73 20 4e 54 20 31 30 2e 30 3b 20 57	ndows NT; 10.0; W				
00c0	69 6e 36 34 30 20 78 36 34 29 20 41 70 6c 65	in64; x64) Apple				
00d0	57 65 62 4b 69 74 2f 35 33 37 2e 33 36 20 28 4b	WebKit/5.37.36 (K				
00e0	48 54 4d 4c 2c 20 6c 69 6b 65 20 47 65 63 6b 6f	HTML, li ke Gecko				
00f0	29 24 43 68 72 6f 65 2f 31 30 32 3e 2e 30 2e 35) Chrome /102.0.5				



PASSO 3 - Capture a tela desses 3 pacotes expandidos no Wireshark, mostrando todos campos de endereço IP Origem e Destino. Enfatize nessas capturas de tela as seguintes informações do TCP para cada um dos três segmentos: Source Port, Destination Port, Sequence Number e Acknowledgment Number.

<<<<<<<<cole aqui as três capturas de telas>>>>>>>>>>>>>>



Wi-Fi

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

ip.adr == 128.119.245.12

No. Time Source Destination Protocol Length Info

303 4.738533 192.168.0.104 128.119.245.12 TCP 66 59915 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1

304 4.741105 128.119.245.12 TCP 66 59915 → 80 [SYN ACK] Seq=1 Ack=1 Win=1412 SACK_PERM=1 WS=128

315 4.871729 128.119.245.12 TCP 66 59915 → 80 [ACK] Seq=1 Ack=2 Win=1412 SACK_PERM=1 WS=128

316 4.871970 128.119.245.12 TCP 66 59915 → 80 [ACK] Seq=1 Ack=2 Win=1412 SACK_PERM=1 WS=128

317 4.872764 128.119.245.12 TCP 66 59915 → 80 [ACK] Seq=1 Ack=2 Win=1412 SACK_PERM=1 WS=128

318 4.883654 128.119.245.12 TCP 66 59915 → 80 [ACK] Seq=1 Ack=2 Win=1412 SACK_PERM=1 WS=128

319 4.883777 128.119.245.12 TCP 66 59915 → 80 [ACK] Seq=1 Ack=2 Win=1412 SACK_PERM=1 WS=128

335 5.004240 128.119.245.12 TCP 66 59915 → 80 [ACK] Seq=1 Ack=2 Win=1412 SACK_PERM=1 WS=128

336 5.005758 128.119.245.12 TCP 66 59915 → 80 [ACK] Seq=1 Ack=2 Win=1412 SACK_PERM=1 WS=128

337 5.005758 128.119.245.12 TCP 66 59915 → 80 [ACK] Seq=1 Ack=2 Win=1412 SACK_PERM=1 WS=128

338 5.005896 128.119.245.12 TCP 66 59915 → 80 [ACK] Seq=1 Ack=2 Win=1412 SACK_PERM=1 WS=128

339 5.006008 128.119.245.12 TCP 66 59915 → 80 [ACK] Seq=1 Ack=2 Win=1412 SACK_PERM=1 WS=128

350 5.169752 128.119.245.12 TCP 66 59915 → 80 [ACK] Seq=1 Ack=2 Win=1412 SACK_PERM=1 WS=128

351 5.170102 128.119.245.12 TCP 66 59915 → 80 [ACK] Seq=1 Ack=2 Win=1412 SACK_PERM=1 WS=128

352 5.193921 128.119.245.12 TCP 66 59915 → 80 [ACK] Seq=1 Ack=2 Win=1412 SACK_PERM=1 WS=128

368 5.326462 128.119.245.12 TCP 66 59915 → 80 [ACK] Seq=1 Ack=2 Win=1412 SACK_PERM=1 WS=128

369 5.327753 128.119.245.12 TCP 66 59915 → 80 [ACK] Seq=1 Ack=2 Win=1412 SACK_PERM=1 WS=128

370 5.327753 128.119.245.12 TCP 66 59915 → 80 [ACK] Seq=1 Ack=2 Win=1412 SACK_PERM=1 WS=128

371 5.327824 128.119.245.12 TCP 66 59915 → 80 [ACK] Seq=1 Ack=2 Win=1412 SACK_PERM=1 WS=128

372 5.329420 128.119.245.12 TCP 66 59915 → 80 [ACK] Seq=1 Ack=2 Win=1412 SACK_PERM=1 WS=128

373 5.329541 128.119.245.12 TCP 66 59915 → 80 [ACK] Seq=1 Ack=2 Win=1412 SACK_PERM=1 WS=128

374 5.329660 128.119.245.12 TCP 66 59915 → 80 [ACK] Seq=1 Ack=2 Win=1412 SACK_PERM=1 WS=128

1000 = Header Length: 20 bytes (5)

Flags: 0x012 (S-Ack)

0000 da f3 08 27 fd c1 50 c7 bf f7 d5 ea 08 00 45 00 ...P.....E:

0010 00 34 00 00 40 00 2e 06 16 30 08 77 f5 0c c0 a8 4..@..0w...

0020 00 68 00 50 ea 0b 2f 8e 1b fc 7f 6d c6 e1 88 12 h..P...m....

0030 00 30 72 10 4a 57 00 00 02 04 05 84 01 01 04 02 01 03 ..7M.....

0040 03 07 ..

Flags (12 bits) (tcp.flags), 2 byte(s)

26°C Chuva fraca

22:02 01/06/2022

Wi-Fi

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

ip.adr == 128.119.245.12

No. Time Source Destination Protocol Length Info

303 4.738533 192.168.0.104 128.119.245.12 TCP 66 59915 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1

304 4.741105 128.119.245.12 TCP 66 59915 → 80 [SYN ACK] Seq=1 Ack=1 Win=1412 SACK_PERM=1 WS=128

315 4.871729 128.119.245.12 TCP 66 59915 → 80 [ACK] Seq=1 Ack=2 Win=1412 SACK_PERM=1 WS=128

316 4.871970 128.119.245.12 TCP 66 59915 → 80 [ACK] Seq=1 Ack=2 Win=1412 SACK_PERM=1 WS=128

317 4.872764 128.119.245.12 TCP 66 59915 → 80 [ACK] Seq=1 Ack=2 Win=1412 SACK_PERM=1 WS=128

318 4.883654 128.119.245.12 TCP 66 59915 → 80 [ACK] Seq=1 Ack=2 Win=1412 SACK_PERM=1 WS=128

319 4.883777 128.119.245.12 TCP 66 59915 → 80 [ACK] Seq=1 Ack=2 Win=1412 SACK_PERM=1 WS=128

335 5.004240 128.119.245.12 TCP 66 59915 → 80 [ACK] Seq=1 Ack=2 Win=1412 SACK_PERM=1 WS=128

336 5.005758 128.119.245.12 TCP 66 59915 → 80 [ACK] Seq=1 Ack=2 Win=1412 SACK_PERM=1 WS=128

337 5.005758 128.119.245.12 TCP 66 59915 → 80 [ACK] Seq=1 Ack=2 Win=1412 SACK_PERM=1 WS=128

338 5.005896 128.119.245.12 TCP 66 59915 → 80 [ACK] Seq=1 Ack=2 Win=1412 SACK_PERM=1 WS=128

339 5.006008 128.119.245.12 TCP 66 59915 → 80 [ACK] Seq=1 Ack=2 Win=1412 SACK_PERM=1 WS=128

350 5.169752 128.119.245.12 TCP 66 59915 → 80 [ACK] Seq=1 Ack=2 Win=1412 SACK_PERM=1 WS=128

351 5.170102 128.119.245.12 TCP 66 59915 → 80 [ACK] Seq=1 Ack=2 Win=1412 SACK_PERM=1 WS=128

352 5.193921 128.119.245.12 TCP 66 59915 → 80 [ACK] Seq=1 Ack=2 Win=1412 SACK_PERM=1 WS=128

368 5.326462 128.119.245.12 TCP 66 59915 → 80 [ACK] Seq=1 Ack=2 Win=1412 SACK_PERM=1 WS=128

369 5.327753 128.119.245.12 TCP 66 59915 → 80 [ACK] Seq=1 Ack=2 Win=1412 SACK_PERM=1 WS=128

370 5.327753 128.119.245.12 TCP 66 59915 → 80 [ACK] Seq=1 Ack=2 Win=1412 SACK_PERM=1 WS=128

371 5.327824 128.119.245.12 TCP 66 59915 → 80 [ACK] Seq=1 Ack=2 Win=1412 SACK_PERM=1 WS=128

372 5.329420 128.119.245.12 TCP 66 59915 → 80 [ACK] Seq=1 Ack=2 Win=1412 SACK_PERM=1 WS=128

373 5.329541 128.119.245.12 TCP 66 59915 → 80 [ACK] Seq=1 Ack=2 Win=1412 SACK_PERM=1 WS=128

374 5.329660 128.119.245.12 TCP 66 59915 → 80 [ACK] Seq=1 Ack=2 Win=1412 SACK_PERM=1 WS=128

1001 = Header Length: 20 bytes (5)

Flags: 0x010 (Ack)

0000 50 c7 bf f7 d5 ea da f3 08 27 fd c1 08 00 45 00 P.....E:

0010 00 28 aa fa 40 00 80 06 19 41 c0 a8 00 68 00 77 (@..A..h.w

0020 f5 0c ea 05 00 50 7f 6d c6 e1 2f 8e 1b fd 50 16 ..P...m...P

0030 02 00 fb 09 00 00 ..

Flags (12 bits) (tcp.flags), 2 byte(s)

26°C Chuva fraca

22:02 01/06/2022

PASSO 4 - Preencha os campos abaixo usando as informações dos pacotes capturados.

Primeiro Segmento TCP	Segundo Segmento TCP	Terceiro Segmento TCP
Source port: 59915	Source port: 59916	Source port: 80
Destination port: 80	Destination port: 80	Destination port: 59915
Sequence Number: 0	Sequence Number: 0	Sequence Number: 0
ACK Number: 0	ACK Number: 0	ACK Number: 1
Marque com um X as flags que foram habilitadas:	Marque com um X as flags que foram habilitadas:	Marque com um X as flags que foram habilitadas:
<input type="checkbox"/> ACK	<input type="checkbox"/> ACK	X ACK
<input type="checkbox"/> Push	<input type="checkbox"/> Push	<input type="checkbox"/> Push
X Syn	X Syn	X Syn
<input type="checkbox"/> Fin	<input type="checkbox"/> Fin	<input type="checkbox"/> Fin

2 QUESTÃO

Vamos investigar o comportamento do protocolo TCP analisando um rastro dos segmentos TCP enviados e recebidos na transferência de um arquivo de 150KB (contendo o texto de Lewis Carroll's – Aventuras de Alice no País das Maravilhas) do seu computador para um servidor remoto.

1. Capturando uma transferência TCP em massa do seu computador para um servidor remoto

Antes de começar nossa exploração do TCP, precisamos usar o Wireshark para obter um rastreamento de pacotes da transferência TCP de um arquivo do computador para um servidor remoto. Você fará isso acessando uma página da Web que permitirá que você digite o nome de um arquivo armazenado em seu computador (que contém o texto ASCII de Alice no País das Maravilhas) e, em seguida, transfira o arquivo para um servidor Web usando o HTTP POST.

Estamos usando o método POST em vez do método GET, pois gostaríamos de transferir uma grande quantidade de dados do computador para outro computador.

Faça o seguinte:

- Inicie o seu navegador web. Vá para <http://gaia.cs.umass.edu/wireshark-labs/alice.txt> e recupere uma cópia ASCII de Alice no País das Maravilhas. Armazene este arquivo em algum lugar no seu computador.
- Em seguida, vá para <http://gaia.cs.umass.edu/wireshark-labs/TCP-wireshark-file1.html>.

Upload page for TCP Wireshark Lab
Computer Networking: A Top Down Approach, 6th edition
Copyright 2012 J.F. Kurose and K.W. Ross, All Rights Reserved

If you have followed the instructions for the TCP Wireshark Lab, you have *already* downloaded an ASCII copy of Alice and Wonderland from <http://gaia.cs.umass.edu/wireshark-labs/alice.txt> and you also *already* have the Wireshark packet sniffer running and capturing packets on your computer.

Click on the Browse button below to select the directory/file name for the copy of alice.txt that is stored on your computer.

Nenhum arquivo escolhido

Once you have selected the file, click on the "Upload alice.txt file" button below. This will cause your browser to send a copy of alice.txt over an HTTP connection (using TCP) to the web server at gaia.cs.umass.edu. After clicking on the button, wait until a short message is displayed indicating the the upload is complete. Then stop your Wireshark packet sniffer - you're ready to begin analyzing the TCP transfer of alice.txt from your computer to gaia.cs.umass.edu!!

Figura 1: Página para fazer upload do arquivo alice.txt do seu computador para gaia.cs.umass.edu

- Na página, utilize o botão '*'Escolher arquivo'*' deste formulário para introduzir o nome do arquivo (nome completo do caminho) no computador que contém Alice no País das Maravilhas (ou fazê-lo manualmente). Não pressione o botão "Carregar arquivo alice.txt".
- Agora inicie o Wireshark e comece a captura de pacotes (Capture-> Start) e pressione OK na tela Wireshark Packet Capture Options (não precisaremos selecionar nenhuma opção aqui).
- Voltando ao seu navegador, pressione o botão "Upload alice.txt file" para carregar o arquivo para o servidor gaia.cs.umass.edu. Uma vez que o arquivo foi carregado, uma breve mensagem de parabéns será exibida na janela do navegador.
- Pare a captura de pacotes do Wireshark.

2. Um primeiro olhar para o trace capturado

Antes de analisar detalhadamente o comportamento da conexão TCP, vamos dar uma visão geral do rastreamento. Vamos começar examinando a mensagem HTTP POST que carregou o arquivo alice.txt para gaia.cs.umass.edu do seu computador. Encontre esse arquivo em seu rastreamento do Wireshark e expanda a mensagem HTTP para que possamos dar uma olhada na mensagem HTTP POST com mais cuidado. Sua tela do Wireshark deve se parecer com a Figura 2.

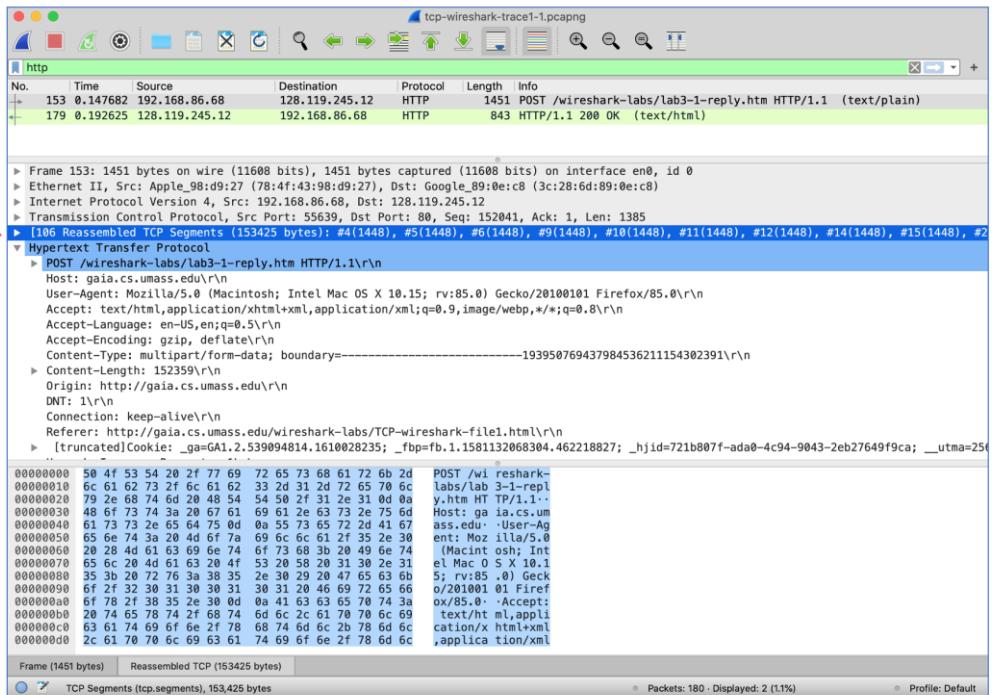
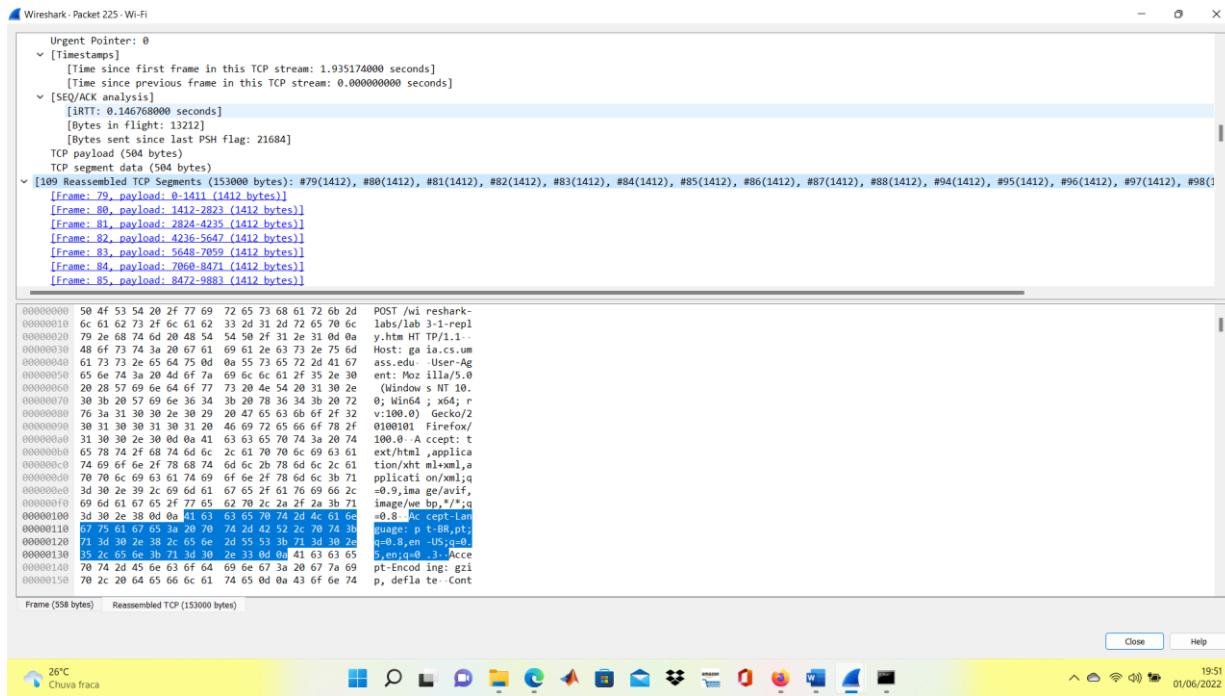


Figura 2: Expandindo a mensagem HTTP POST que carregou alice.txt do seu computador para gaia.cs.umass.edu

Há algumas coisas a serem observadas aqui:

O corpo da mensagem HTTP POST da camada de aplicativo contém o conteúdo do arquivo alice.txt, que é um arquivo grande com mais de 152 K bytes. OK – não é tão grande, mas vai ser muito grande para esta mensagem HTTP POST estar contida em apenas um segmento TCP!

De fato, conforme mostrado na janela do Wireshark na Figura 2, vemos que a mensagem HTTP POST foi espalhada por 106 segmentos TCP. Isso é mostrado onde a seta vermelha é colocada na Figura 3 [Aparte: o Wireshark não tem uma seta vermelha assim; nós o adicionamos à figura para ser útil ☺]. Se você olhar com mais cuidado, verá que o Wireshark também está sendo muito útil para você, informando que o primeiro segmento TCP que contém o início da mensagem POST é o pacote nº 4 no rastreamento específico para o exemplo da Figura 2 , que é o rastreamento tcp-wireshark-trace1-1 observado na nota de rodapé 2. O segundo segmento TCP contendo a mensagem POST no pacote #5 no rastreamento e assim por diante.



Vamos agora “sujar as mãos” olhando alguns segmentos TCP.

Primeiro, filtре os pacotes exibidos na janela do Wireshark digitando “tcp” (minúsculas, sem aspas, e não se esqueça de pressionar o retorno após inserir!) Sua tela do Wireshark deve parecer com a Figura 4. Na Figura 4, observamos o segmento TCP que tem seu bit SYN definido – esta é a primeira mensagem TCP no handshake de três vias que configura a conexão TCP para gaia.cs .umass.edu que eventualmente carregará a mensagem HTTP POST e o arquivo alice.txt. Também notamos o segmento SYNACK (a segunda etapa do handshake de três vias TCP), bem como o segmento TCP (pacote nº 4, conforme discutido acima) que carrega a mensagem POST e o início do arquivo alice.txt. Claro, se você estiver pegando seu próprio arquivo de rastreamento, os números dos pacotes serão diferentes, mas você deverá ver um comportamento semelhante ao mostrado nas Figuras 2 e 3.

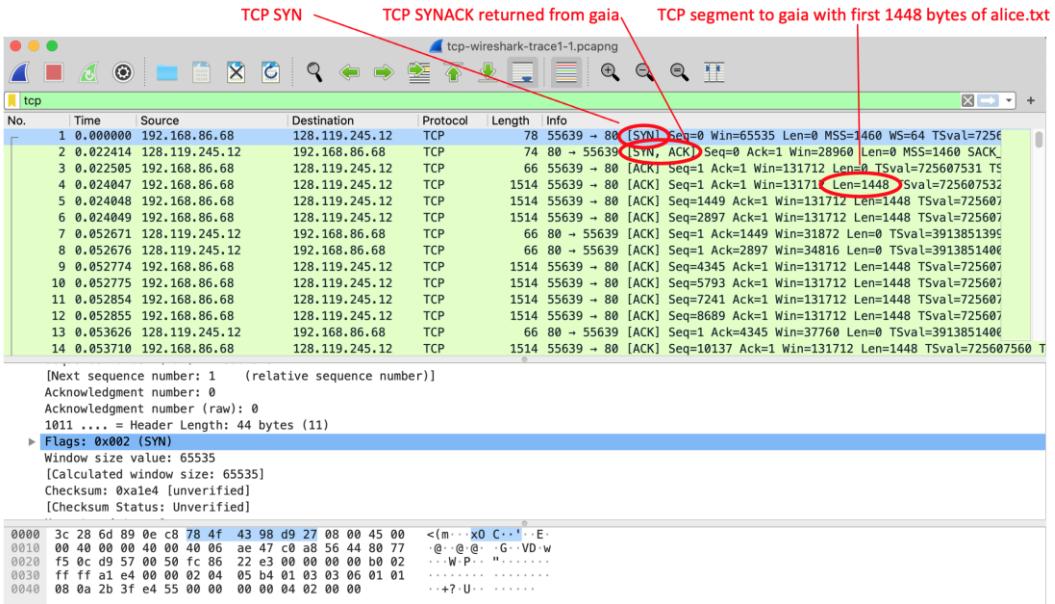
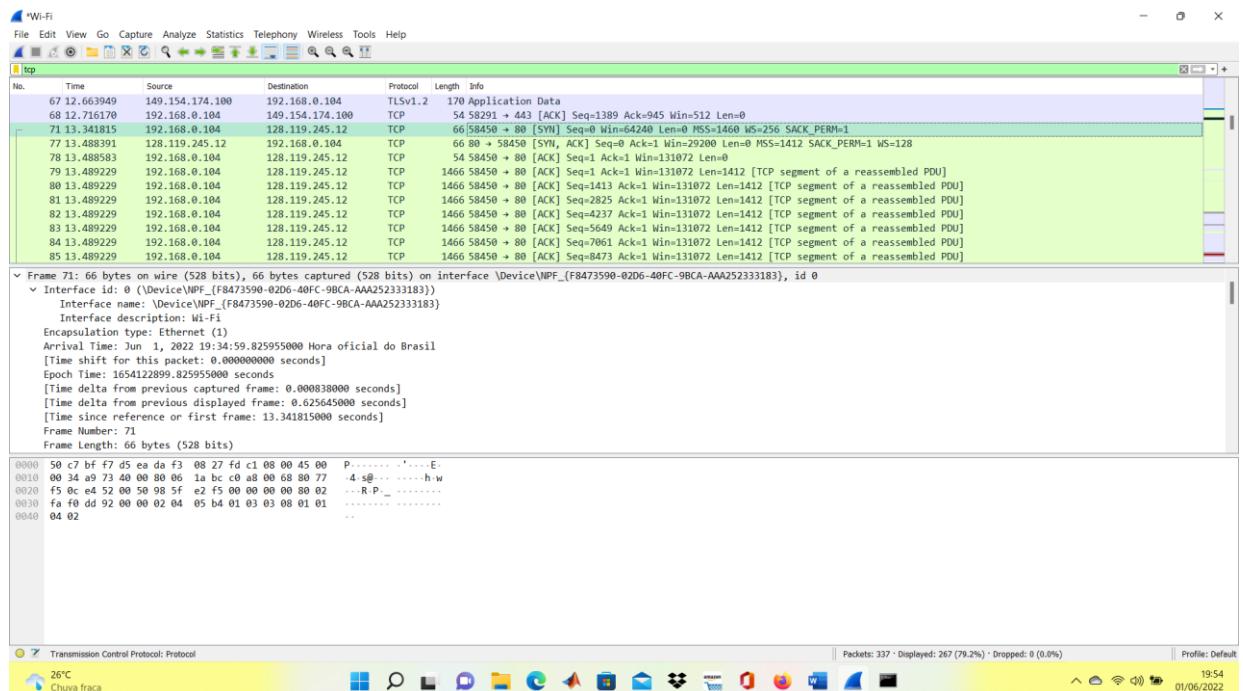


Figura 2: Segmentos TCP envolvidos no envio da mensagem HTTP POST (incluindo o arquivo alice.txt) para gaia.cs.umass.edu



Antes de analisar o comportamento da conexão TCP em detalhes, filtre os pacotes exibidos na janela do Wireshark digitando "tcp" na janela de especificação do filtro de exibição na parte superior da janela do Wireshark.

Pergunta: Você deve ver é uma série de mensagens TCP e HTTP entre seu computador e gaia.cs.umass.edu. Você deve ver o handshake inicial de três vias contendo uma mensagem SYN. Você deve ver uma mensagem HTTP POST. Além disso, o que há de diferente da primeira experiência acima?

Responda às seguintes perguntas:

- Qual é o endereço IP e o número de porta TCP usado pelo computador cliente (origem) que está transferindo o arquivo para gaia.cs.umass.edu?

Internet Protocol Version 4, Src: 192.168.0.104, Dst: 128.119.245.12

- Qual é o endereço IP de gaia.cs.umass.edu? Em que número de porta está enviando e recebendo segmentos TCP para essa conexão?

Transmission Control Protocol, Src Port: 58450, Dst Port: 80, Seq: 152497, Ack: 1, Len: 504

IP: 128.119.245.12 , porta: 80

- Qual é o endereço IP e o número da porta TCP usado pelo computador cliente (origem) para transferir o arquivo para gaia.cs.umass.edu?

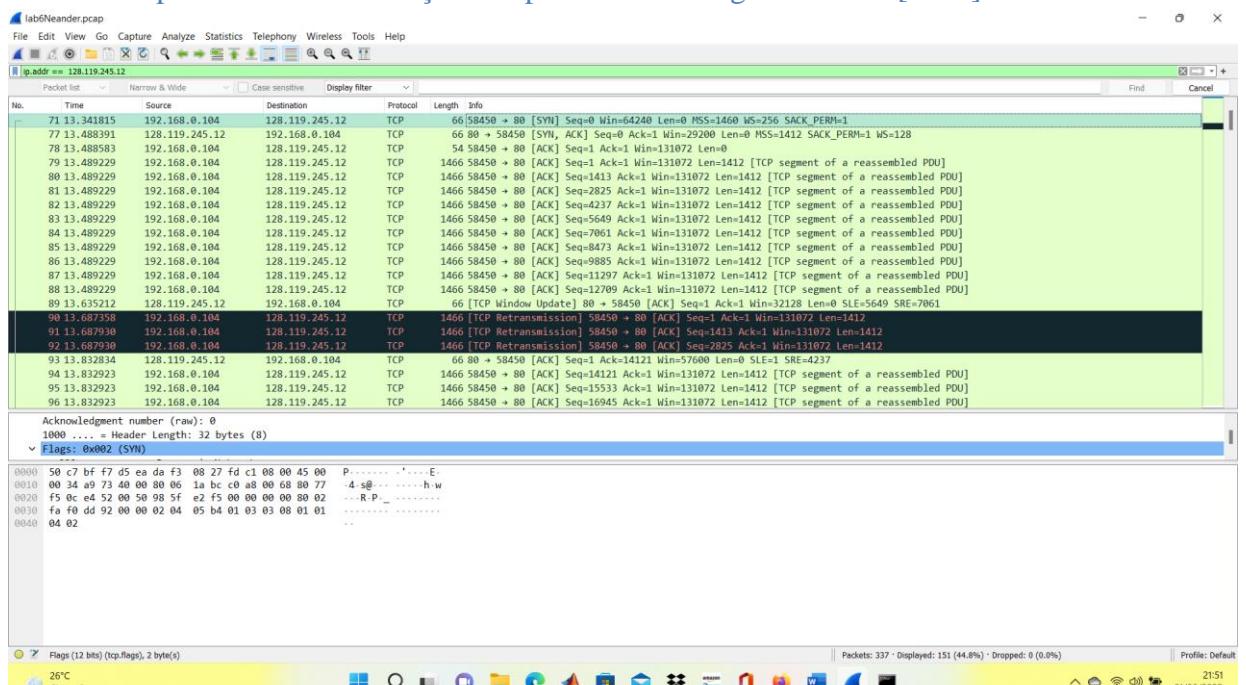
IP: 192.168.0.104, porta: 58450

Nota: Uma vez que este laboratório é sobre TCP em vez de HTTP, vamos mudar a janela "listagem de pacotes capturados" do Wireshark para que mostre informações sobre os segmentos TCP contendo as mensagens HTTP, em vez de sobre as mensagens HTTP. Para que o Wireshark faça isso, selecione Analisar-> Protocolos habilitados. Em seguida, desmarque a caixa HTTP e selecione OK.

- Qual é o número de sequência do segmento TCP SYN que é usado para iniciar a conexão TCP entre o computador cliente e gaia.cs.umass.edu? O que é no segmento que identifica o segmento como um segmento SYN?

Sequence Number: 0 (relative sequence number)

Há um campo dentre as informações do pacote com o seguinte dado: [SYN]



- Qual é o número de sequência do segmento SYNACK enviado por gaia.cs.umass.edu para o computador cliente em resposta ao SYN? Qual é o valor do campo

Reconhecimento no segmento SYNACK? Como gaia.cs.umass.edu determinou esse valor? O que é no segmento que identifica o segmento como um segmento SYNACK?

O campo reconhecimento (**Acknowledgment**) tem valor 1.

Sequence Number: 0 (relative sequence number)

O segment é identificado pelas flags.

Flags: 0x012 (SYN, 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 = **Acknowledgment**: Set

.... 0... = Push: Not set

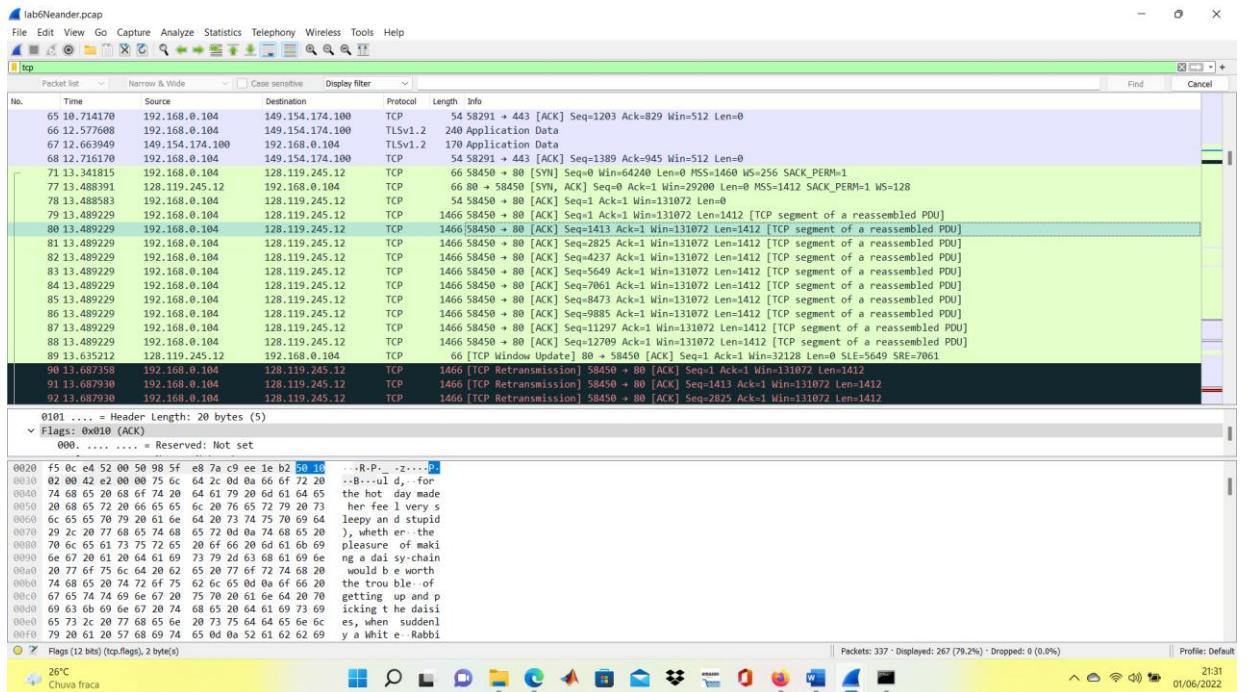
....0.. = Reset: Not set

....1. = Syn: Set

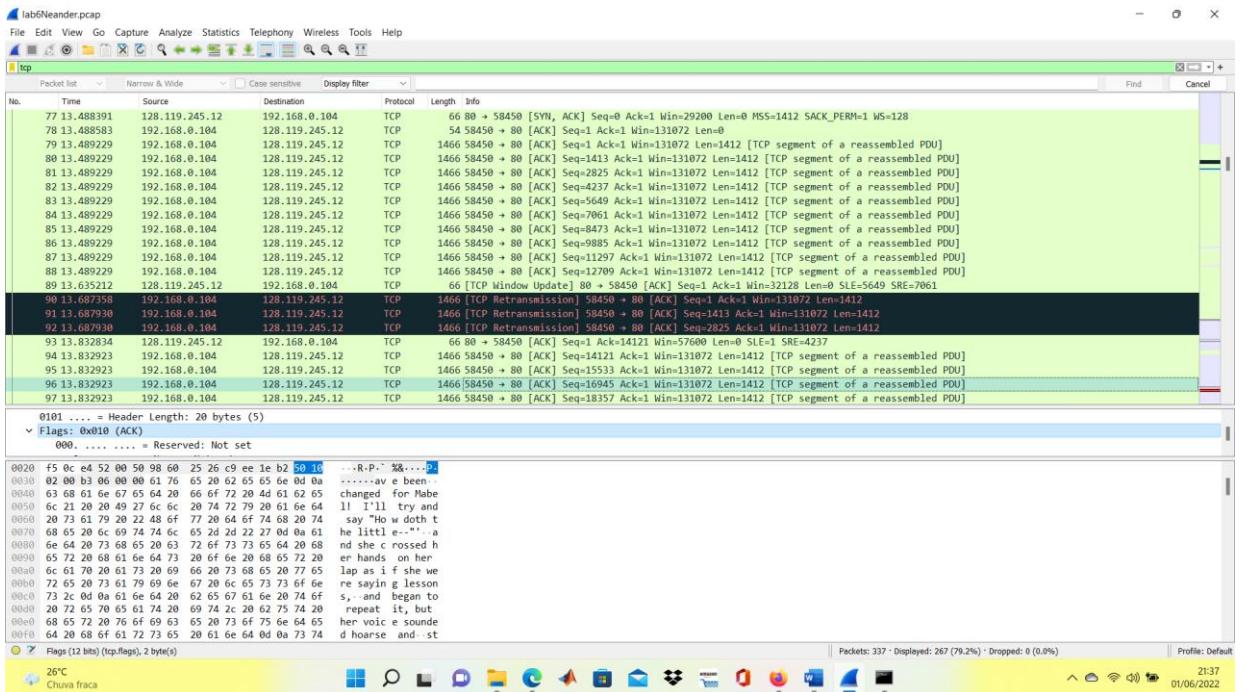
6. Qual é o número de sequência do segmento TCP que contém o comando HTTP POST? Observe que, para encontrar o comando POST, você precisará digitar no campo de conteúdo de pacote na parte inferior da janela Wireshark, procurando um segmento com um "POST" dentro de seu campo DATA.

Sequence Number: 152497 (relative sequence number)

7. Considere o segmento TCP contendo o HTTP POST como o primeiro segmento na conexão TCP. Quais são os números de sequência dos primeiros seis segmentos na conexão TCP (incluindo o segmento que contém o HTTP POST)? Em que horário foi enviado cada segmento? Quando foi recebido o ACK para cada segmento?



- Sequence Number: 1 (relative sequence number) Arrival Time: Jun 1, 2022 19:34:59.973369000 Hora oficial do Brasil
 - Sequence Number: 1413 (relative sequence number) Arrival Time: Jun 1, 2022 19:34:59.973369000 Hora oficial do Brasil
 - Sequence Number: 2825 (relative sequence number) Arrival Time: Jun 1, 2022 19:34:59.973369000 Hora oficial do Brasil
 - Sequence Number: 4237 (relative sequence number) Arrival Time: Jun 1, 2022 19:34:59.973369000 Hora oficial do Brasil
 - Sequence Number: 5649 (relative sequence number) Arrival Time: Jun 1, 2022 19:34:59.973369000 Hora oficial do Brasil
 - Sequence Number: 7061 (relative sequence number) Arrival Time: Jun 1, 2022 19:34:59.973369000 Hora oficial do Brasil
8. Qual é o comprimento de cada um dos seis primeiros segmentos TCP?
Todos tem Len=1412 bytes de comprimento.
9. Existem segmentos retransmitidos no ficheiro de rastreio? O que você verificou (no rastro) para responder a esta pergunta?
 Há mensagens com campos [TCP Retransmission], supõe-se que esses pacotes foram reenviados.



Nota: Veja sobre o RTT: Wireshark tem um recurso interessante que permite traçar o RTT para cada um dos segmentos TCP enviados. Selecione um segmento TCP na janela "lista de pacotes capturados" que está sendo enviada do cliente para o servidor gaia.cs.umass.edu. Em seguida, selecione: Statistics-> TCP Stream Graph-> Round Trip Time Graph.

OBS: NÃO ESQUEÇA DE ENVIAR O LAB

