C 0 192.168.56.101

Como vai ese protocolo http??

Páxina Inicio

Formulario GET Formulario POST

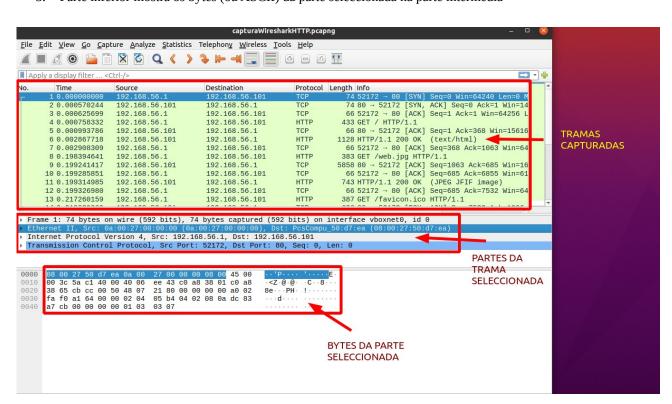
Introdución a Wireshark. Práctica guiada 1.1

Veremos co Wireshark os paquetes intercambiados entre un navegador web (nun equipo cliente), e o servidor, empregando o protocolo http. Neste caso o servidor e o cliente están no mesmo segmento da rede, non hai ningún rooter entre eles. O que pode ver o cliente é unha páxina web (en formato html):

Wireshark (capturaHTTP.pcapng)

Wireshark é un **capturador** e **analizador** de paquetes de rede, que nos permite analizar as tramas de rede. A súa ventana está dividida en 3 partes:

- 1. Parte superior mostra todas as tramas
- 2. Parte intermedia mostra as partes da trama seleccionada na parte superior
- 3. Parte inferior mostra os bytes (ou ASCII) da parte seleccionada na parte intermedia



A captura anterior corresponde con un intercambio entre un cliente web e un servidor web, empregando o protocolo http. Está seleccionada:

- 1) A primeira trama, do protocolo TCP. Podes comprobar na parte inferior do teu Wireshark que son 74 bytes.
- 2) As súas partes serán Ethernet, IP e TCP, como pode ver na parte intermedia.
- 3) Se seleccionamos a parte Ethernet II, quedan seleccionados os 14 bytes que corresponden á cabeceira Ethernet: as direccións MAC do equipo orixe e equipo destino: **08:00:27:50:d7:ea** e **0a:00:27:00:00:00** e 2 bytes para o tipo de trama (IPv4)

Podes ver tamén que o cliente ten a IP 192.168.56.1, e o destino (SERVIDOR) a 192.168.56.101

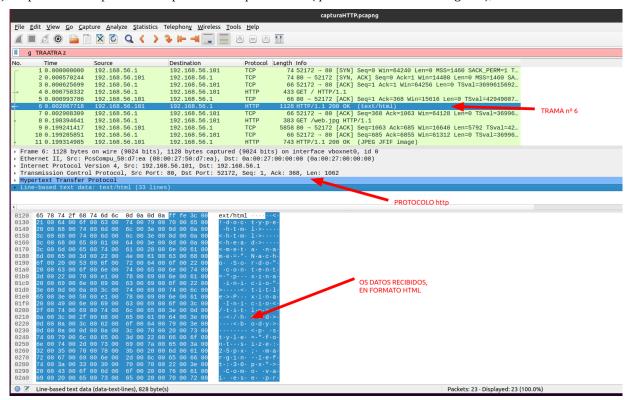
TRAMA 4

A trama 4 é a petición HTTP na cal o cliente solicita a páxina web ao servidor. Pincha sobre ela e comproba que as MAC orixe, MAC destino , e as IP's son as mesmas que as da primeira trama.

Trama 6

Selecciona agora a trama 6, que corresponde ao protocolo HTTP, e é a trama RESPOSTA do servidor. Tes que poder ver:

- 1) A trama nº 6, do protocolo HTTP. Podes comprobar na parte inferior do teu Wireshark que son 828 bytes.
- 2) As súas partes serán Ethernet, IP, TCP e HTTP (Hypertext Transfer Protocol)
- 3) Na parte inferior podes ver na parte dereita que os datos (que se mostrarían no navegador), son HTML



Poderíamos ver na parte intermedia de novo as direccións do servidor e do cliente:

- 1. Ethernet: o servidor que envía ten a MAC 08:00:27:50:d7:ea, o cliente a MAC 0a:00:27:00:00
- 2. **IP:** O servidor envía desde a IP 192.168.56.101, e o cliente recibe na súa IP 192.168.56.1
- 3. TCP: O servidor emprega o porto 80, que é un número de porto reservado para o protocolo HTTP

Trama 8

Correspóndese coa petición http de unha imaxe, enviada do cliente ao servidor. Identifica de novo direccións orixe e cliente para:

- 1. Ethernet:
- 2. IP:
- 3. TCP:

Trama 11

Correspóndese coa resposta do servidor, enviando unha imaxe jpeg.

Identifica de novo as direccións que corresponden a ese trama:

- 1. Ethernet:
- 2. **IP**:
- 3. **TCP**:
- 4. **HTTP**: Busca dentro da parte intermedia que os datos son unha imaxe jpeg, como podes ver na imaxe seguinte:

```
Transmission Control Protocol, Src Port: 80, Dst Port: 52172, Seq: 6855, Ack: 685, Len: 677

[2 Reassembled TCP Segments (6469 bytes): #9(5792), #11(677)]

Hypertext Transfer Protocol

HTTP/1.1 200 OK\r\n

Date: Tue, 04 Oct 2022 15:31:05 GMT\r\n

Server: Apache/2.4.3 (Unix) OpenSSL/1.0.1c PHP/5.4.7\r\n

Last-Modified: Tue, 04 Jun 2013 22:28:49 GMT\r\n

ETag: "1810-4de5b9ae60a40"\r\n

Accept-Ranges: bytes\r\n

Content-Length: 6160\r\n

Keep-Alive: timeout=5, max=99\r\n

Connection: Keep-Alive\r\n

Content-Type: image/jpeg\r\n
\r\n

[HTTP response 2/3]

[Time since request: 0.000920344 seconds]

[Prev request in frame: 4]

[Prev response in frame: 6]

[Request in frame: 8]

[Next request in frame: 22]

[Request URI: http://192.168.56.101/ravicon.ico]

File Data: 6160 bytes

**JPEG File Interchange Format*

Marker: Start of Image (0xffd8)

**Marker segment: Befine quantization table(s) (0xFFDB)

**Start of Frame header: Start of Frame (non-differential, Huffman coding) - Baseline DCT (0xFFC0)

**Marker segment: Define Huffman table(s) (0xFFC4)

**Marker segment: Define Huffman table(s) (0xFFC4)

**Marker segment: Define Huffman table(s) (0xFFC4)
```