

Redes de Computador

Fase 1



ISEL

ADEETC

Área Departamental de
Engenharia Electrónica e
Telecomunicações e
de Computadores

Mariana Oliveira 42355

Tomás Carvalho 42357

Leim 41D

Docente: Nuno Cruz

Data: 01/04/2019

Índice

| | |
|-----------------------------|---|
| Objetivo | 1 |
| Desenvolvimento | 2 |
| Explicação de headers | 4 |

Índice de Imagens

| | |
|---------------------------------------|---|
| Figura 1 – XAMPP dashboard PC1 | 2 |
| Figura 2 - PC1 acesso localhost | 2 |
| Figura 3 - Terminal PC1 | 3 |
| Figura 4 - PC2 acesso PC1 | 3 |
| Figura 5 - Packet Capture PC1 | 4 |
| Figura 6 - Packet Capture PC2 | 4 |
| Figura 7 - Tabela de Packets | 6 |

Anexos

| | |
|---|---|
| Template para Headers de Queries HTTP | 6 |
| Template para Headers de Responses HTTP | 7 |
| Exemplo de Headers de Pedido e da resperiva Resposta HTTP | 8 |

Objetivo

Setup e teste de um *webserver* num computador.

Acesso ao servidor por parte de um segundo computador (através do endereço IP do alojador do servidor – computador 1)

Captura e avaliação/explicação dos packets capturados aquando do acesso/ligação ao servidor.

Desenvolvimento

Para este trabalho foi necessário instalar o XAMPP e correr num computador (PC1), no *control panel* do XAMPP foram ativados os vários serviços/módulos do mesmo, verificar *Figura 1*.

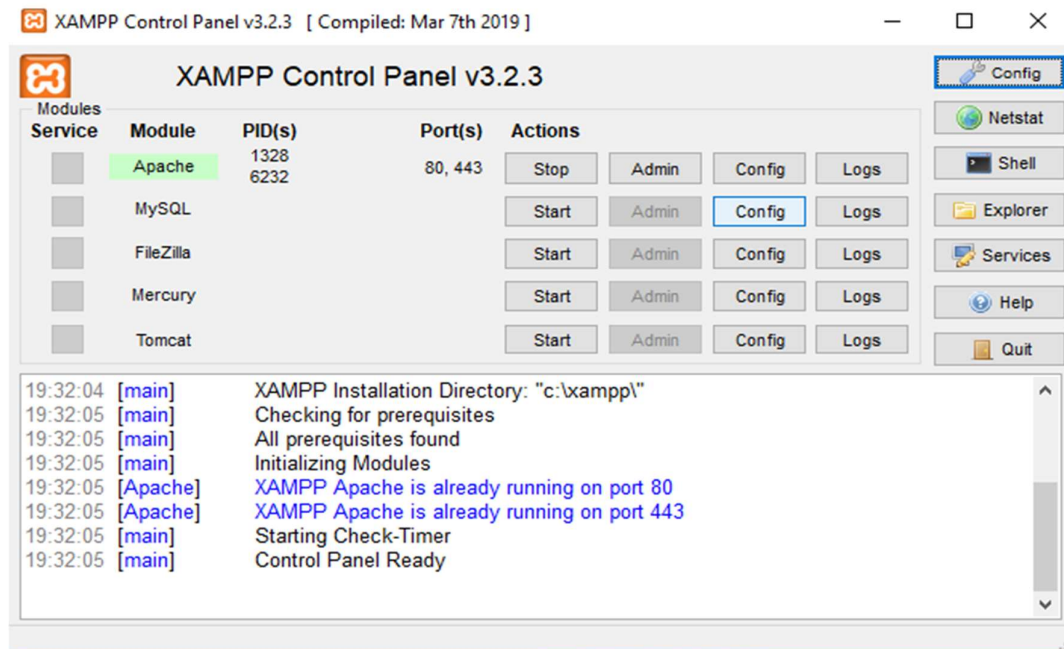


Figura 1 – XAMPP dashboard PC1

De seguida foi feito o teste sobre o servidor do PC1 acedendo a 127.0.0.1, no mesmo.



Figura 2 - PC1 acesso localhost

Confirmado que este estava a correr verificou-se o endereço IP do PC1 no terminal com o comando `ipconfig`, irá aparecer algo semelhante a *Figura 3*.

```
Command Prompt
Connection-specific DNS Suffix . : 
Link-local IPv6 Address . . . . . : fe80::ac78:8028:1148:2283%13
Autoconfiguration IPv4 Address. . : 169.254.34.131
Subnet Mask . . . . . : 255.255.0.0
Default Gateway . . . . . : 

Wireless LAN adapter Local Area Connection* 1:

Media State . . . . . : Media disconnected
Connection-specific DNS Suffix . : 

Wireless LAN adapter Local Area Connection* 3:

Media State . . . . . : Media disconnected
Connection-specific DNS Suffix . : 

Wireless LAN adapter Wi-Fi:

Connection-specific DNS Suffix . : eduroam.local
Link-local IPv6 Address . . . . . : fe80::a562:df70:9c40:3c19%10
IPv4 Address. . . . . : 10.10.37.34
Subnet Mask . . . . . : 255.255.252.0
Default Gateway . . . . . : 10.10.39.254

Ethernet adapter Bluetooth Network Connection:

Media State . . . . . : Media disconnected
Connection-specific DNS Suffix . : 

C:\Users\tomas>
```

Figura 3 - Terminal PC1

No PC2 acedeu-se ao servidor do PC1 através do IP do mesmo e como se pode verificar na *Figura 4* houve ligação.

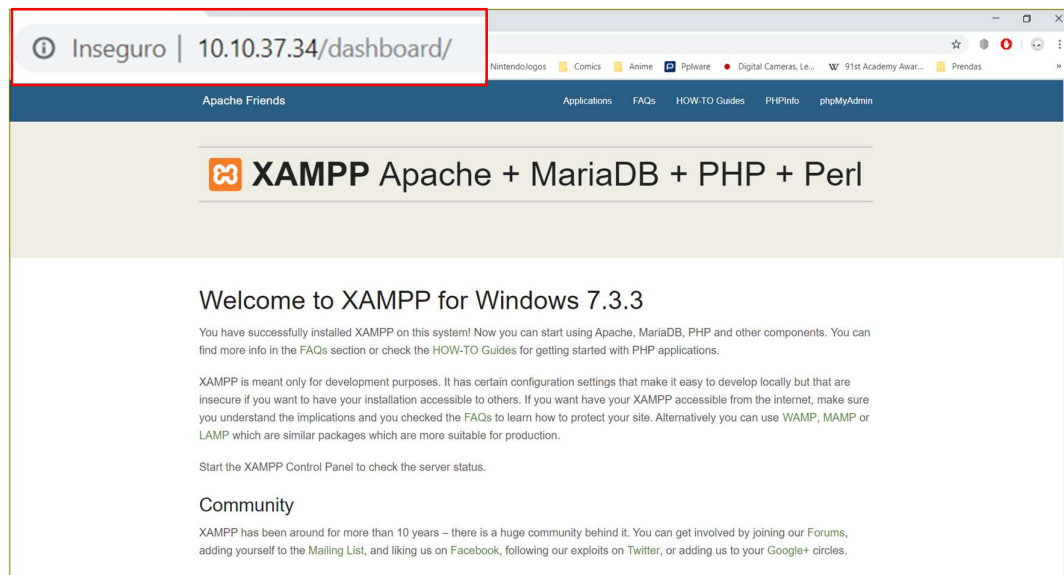
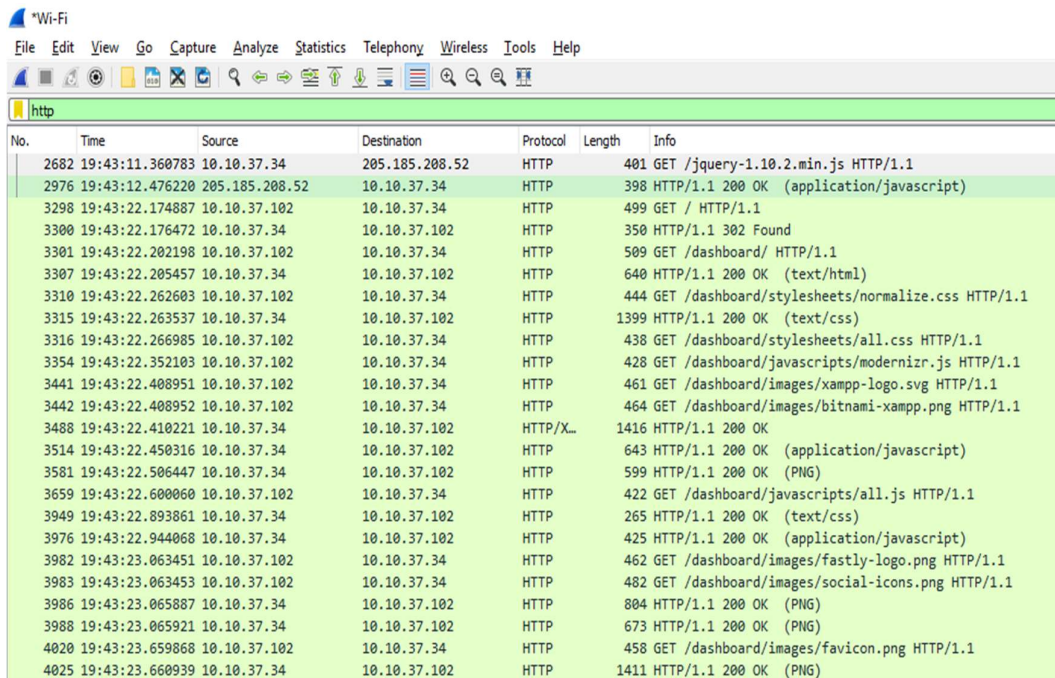


Figura 4 - PC2 acesso PC1

Confirmado o funcionamento do servidor e a ligação entre os computadores, foi repetido o processo com o *wireshark* a correr paralelamente, para fazer a captura de *packets* transmitidos, entre *host* e servidor.

Explicação de headers

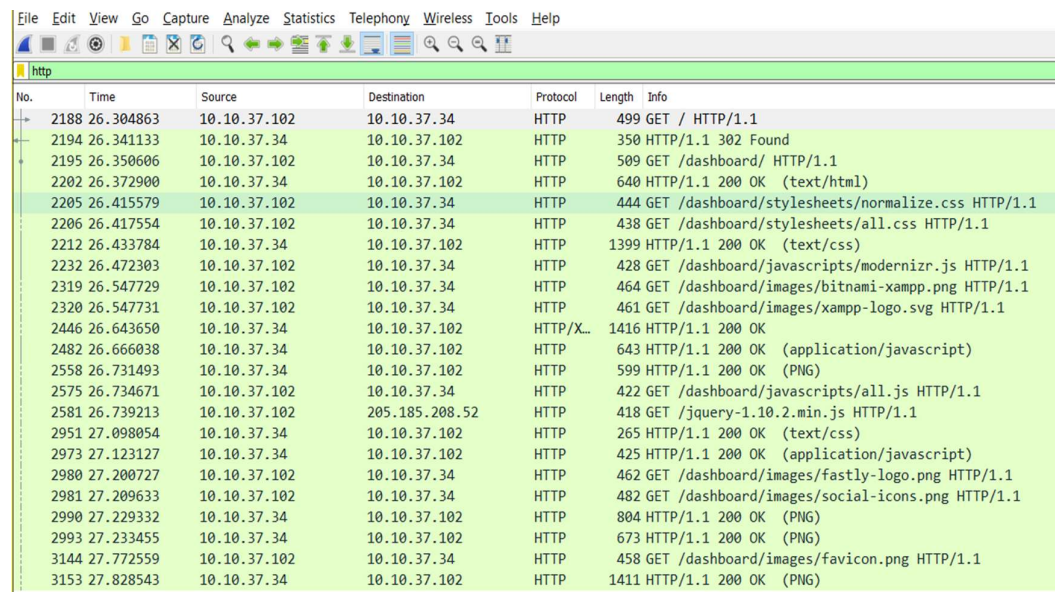
As duas primeiras linhas representadas na *Figura 5* correspondem ao acesso do PC1 ao *localhost* (127.0.0.1). Sendo que a localização (IP) devolve a pagina do servidor é 205.185.208.52. Primeira linha é o *query* e a segunda é o *response*, ou seja, pedido e resposta do acesso.



The image shows a Wireshark packet capture window titled '*Wi-Fi'. The menu bar includes File, Edit, View, Go, Capture, Analyze, Statistics, Telephony, Wireless, Tools, and Help. The packet list pane shows 20 packets. The selected packet is packet 2682, which is an HTTP GET request for /jquery-1.10.2.min.js from 10.10.37.34 to 205.185.208.52. The packet details pane shows the request line: GET /jquery-1.10.2.min.js HTTP/1.1. The packet bytes pane shows the raw data of the request.

| No. | Time | Source | Destination | Protocol | Length | Info |
|------|-----------------|----------------|----------------|----------|--------|---|
| 2682 | 19:43:11.360783 | 10.10.37.34 | 205.185.208.52 | HTTP | 401 | GET /jquery-1.10.2.min.js HTTP/1.1 |
| 2976 | 19:43:12.476220 | 205.185.208.52 | 10.10.37.34 | HTTP | 398 | HTTP/1.1 200 OK (application/javascript) |
| 3298 | 19:43:22.174887 | 10.10.37.102 | 10.10.37.34 | HTTP | 499 | GET / HTTP/1.1 |
| 3300 | 19:43:22.176472 | 10.10.37.34 | 10.10.37.102 | HTTP | 350 | HTTP/1.1 302 Found |
| 3301 | 19:43:22.202198 | 10.10.37.102 | 10.10.37.34 | HTTP | 509 | GET /dashboard/ HTTP/1.1 |
| 3307 | 19:43:22.205457 | 10.10.37.34 | 10.10.37.102 | HTTP | 640 | HTTP/1.1 200 OK (text/html) |
| 3310 | 19:43:22.262603 | 10.10.37.102 | 10.10.37.34 | HTTP | 444 | GET /dashboard/stylesheets/normalize.css HTTP/1.1 |
| 3315 | 19:43:22.263537 | 10.10.37.34 | 10.10.37.102 | HTTP | 1399 | HTTP/1.1 200 OK (text/css) |
| 3316 | 19:43:22.266985 | 10.10.37.102 | 10.10.37.34 | HTTP | 438 | GET /dashboard/stylesheets/all.css HTTP/1.1 |
| 3354 | 19:43:22.352103 | 10.10.37.102 | 10.10.37.34 | HTTP | 428 | GET /dashboard/javascripts/modernizr.js HTTP/1.1 |
| 3441 | 19:43:22.408951 | 10.10.37.102 | 10.10.37.34 | HTTP | 461 | GET /dashboard/images/xampp-logo.svg HTTP/1.1 |
| 3442 | 19:43:22.408952 | 10.10.37.102 | 10.10.37.34 | HTTP | 464 | GET /dashboard/images/bitnami-xampp.png HTTP/1.1 |
| 3488 | 19:43:22.410221 | 10.10.37.102 | 10.10.37.102 | HTTP/X.. | 1416 | HTTP/1.1 200 OK |
| 3514 | 19:43:22.450316 | 10.10.37.34 | 10.10.37.102 | HTTP | 643 | HTTP/1.1 200 OK (application/javascript) |
| 3581 | 19:43:22.506447 | 10.10.37.34 | 10.10.37.102 | HTTP | 599 | HTTP/1.1 200 OK (PNG) |
| 3659 | 19:43:22.600060 | 10.10.37.102 | 10.10.37.34 | HTTP | 422 | GET /dashboard/javascripts/all.js HTTP/1.1 |
| 3949 | 19:43:22.893861 | 10.10.37.34 | 10.10.37.102 | HTTP | 265 | HTTP/1.1 200 OK (text/css) |
| 3976 | 19:43:22.944068 | 10.10.37.34 | 10.10.37.102 | HTTP | 425 | HTTP/1.1 200 OK (application/javascript) |
| 3982 | 19:43:23.063451 | 10.10.37.102 | 10.10.37.34 | HTTP | 462 | GET /dashboard/images/fastly-logo.png HTTP/1.1 |
| 3983 | 19:43:23.063453 | 10.10.37.102 | 10.10.37.34 | HTTP | 482 | GET /dashboard/images/social-icons.png HTTP/1.1 |
| 3986 | 19:43:23.065887 | 10.10.37.34 | 10.10.37.102 | HTTP | 804 | HTTP/1.1 200 OK (PNG) |
| 3988 | 19:43:23.065921 | 10.10.37.34 | 10.10.37.102 | HTTP | 673 | HTTP/1.1 200 OK (PNG) |
| 4020 | 19:43:23.659868 | 10.10.37.102 | 10.10.37.34 | HTTP | 458 | GET /dashboard/images/favicon.png HTTP/1.1 |
| 4025 | 19:43:23.660939 | 10.10.37.34 | 10.10.37.102 | HTTP | 1411 | HTTP/1.1 200 OK (PNG) |

Figura 5 - Packet Capture PC1



The image shows a Wireshark packet capture window titled '*Wi-Fi'. The menu bar includes File, Edit, View, Go, Capture, Analyze, Statistics, Telephony, Wireless, Tools, and Help. The packet list pane shows 20 packets. The selected packet is packet 2188, which is an HTTP GET request for / HTTP/1.1 from 10.10.37.102 to 10.10.37.34. The packet details pane shows the request line: GET / HTTP/1.1. The packet bytes pane shows the raw data of the request.

| No. | Time | Source | Destination | Protocol | Length | Info |
|------|-----------|--------------|----------------|----------|--------|---|
| 2188 | 26.304863 | 10.10.37.102 | 10.10.37.34 | HTTP | 499 | GET / HTTP/1.1 |
| 2194 | 26.341133 | 10.10.37.34 | 10.10.37.102 | HTTP | 350 | HTTP/1.1 302 Found |
| 2195 | 26.350606 | 10.10.37.102 | 10.10.37.34 | HTTP | 509 | GET /dashboard/ HTTP/1.1 |
| 2202 | 26.372900 | 10.10.37.34 | 10.10.37.102 | HTTP | 640 | HTTP/1.1 200 OK (text/html) |
| 2205 | 26.415579 | 10.10.37.102 | 10.10.37.34 | HTTP | 444 | GET /dashboard/stylesheets/normalize.css HTTP/1.1 |
| 2206 | 26.417554 | 10.10.37.102 | 10.10.37.34 | HTTP | 438 | GET /dashboard/stylesheets/all.css HTTP/1.1 |
| 2212 | 26.433784 | 10.10.37.34 | 10.10.37.102 | HTTP | 1399 | HTTP/1.1 200 OK (text/css) |
| 2232 | 26.472303 | 10.10.37.102 | 10.10.37.34 | HTTP | 428 | GET /dashboard/javascripts/modernizr.js HTTP/1.1 |
| 2319 | 26.547729 | 10.10.37.102 | 10.10.37.34 | HTTP | 464 | GET /dashboard/images/bitnami-xampp.png HTTP/1.1 |
| 2320 | 26.547731 | 10.10.37.102 | 10.10.37.34 | HTTP | 461 | GET /dashboard/images/xampp-logo.svg HTTP/1.1 |
| 2446 | 26.643650 | 10.10.37.34 | 10.10.37.102 | HTTP/X.. | 1416 | HTTP/1.1 200 OK |
| 2482 | 26.666038 | 10.10.37.34 | 10.10.37.102 | HTTP | 643 | HTTP/1.1 200 OK (application/javascript) |
| 2558 | 26.731493 | 10.10.37.34 | 10.10.37.102 | HTTP | 599 | HTTP/1.1 200 OK (PNG) |
| 2575 | 26.734671 | 10.10.37.102 | 10.10.37.34 | HTTP | 422 | GET /dashboard/javascripts/all.js HTTP/1.1 |
| 2581 | 26.739213 | 10.10.37.102 | 205.185.208.52 | HTTP | 418 | GET /jquery-1.10.2.min.js HTTP/1.1 |
| 2951 | 27.098054 | 10.10.37.34 | 10.10.37.102 | HTTP | 265 | HTTP/1.1 200 OK (text/css) |
| 2973 | 27.123127 | 10.10.37.34 | 10.10.37.102 | HTTP | 425 | HTTP/1.1 200 OK (application/javascript) |
| 2980 | 27.200727 | 10.10.37.102 | 10.10.37.34 | HTTP | 462 | GET /dashboard/images/fastly-logo.png HTTP/1.1 |
| 2981 | 27.209633 | 10.10.37.102 | 10.10.37.34 | HTTP | 482 | GET /dashboard/images/social-icons.png HTTP/1.1 |
| 2990 | 27.229332 | 10.10.37.34 | 10.10.37.102 | HTTP | 804 | HTTP/1.1 200 OK (PNG) |
| 2993 | 27.233455 | 10.10.37.34 | 10.10.37.102 | HTTP | 673 | HTTP/1.1 200 OK (PNG) |
| 3144 | 27.772559 | 10.10.37.102 | 10.10.37.34 | HTTP | 458 | GET /dashboard/images/favicon.png HTTP/1.1 |
| 3153 | 27.828543 | 10.10.37.34 | 10.10.37.102 | HTTP | 1411 | HTTP/1.1 200 OK (PNG) |

Figura 6 - Packet Capture PC2

Da terceira linha para a frente são listados os packets trocados entre PC1 e PC2. Sendo, portanto, mostrados os mesmos packets em ambos os computadores, se bem que não necessariamente pela mesma ordem – *Figura 6*.

A seguinte tabela contém a explicação organizada dos headers como capturados no PC1 - *Figura 5*.

| Nº linha | Origem | Destino | Protocolo | Status code/ Response Phrase | Request Methode | Tempo de Processamento | Linha de Resposta | Descrição |
|----------|----------------|----------------|-----------|---------------------------------|--------------------|---------------------------|----------------------|--|
| 1 | PC1 | 205.185.208.52 | HTTP | | GET | | 2 | Pedido de acesso ao localhost |
| 2 | 205.185.208.52 | PC1 | HTTP | 200 OK | | 1.1154 | | Permissão de acesso a app JavaScript |
| 3 | PC2 | PC1 | HTTP | | GET | | 4 | Pedido de acesso ao Web Server PC1 |
| 4 | PC1 | PC2 | HTTP | 302 Found | | 0.0015 | | Servidor encontrado |
| 5 | PC2 | PC1 | HTTP | | GET | | 6 | Pedido de acesso a página |
| 6 | PC1 | PC2 | HTTP | 200 OK | | 0.0033 | | Resposta a linha 5 com texto html |
| 7 | PC2 | PC1 | HTTP | | GET | | 8 | Pedido de estilização da página |
| 8 | PC1 | PC2 | HTTP | 200 OK | | 0.0009 | | Resposta a linha 7 com texto css |
| 9 | PC2 | PC1 | HTTP | | GET | | 17 | Pedido de estilização da página |
| 10 | PC2 | PC1 | HTTP | | GET | | 14 | Pedido de acesso a app JavaScript |
| 11 | PC2 | PC1 | HTTP | | GET | | 13 | Pedido de imagem |
| 12 | PC2 | PC1 | HTTP | | GET | | 15 | Pedido de imagem |
| 13 | PC1 | PC2 | HTTP/XML | 200 OK | | 0.0013 | | Resposta a linha 11 SVG |
| 14 | PC1 | PC2 | HTTP | 200 OK | | 0.0982 | | Resposta a linha 10 com app JavaScript |
| 15 | PC1 | PC2 | HTTP | 200 OK | | 0.0975 | | Resposta a linha 12 PNG |
| 16 | PC2 | PC1 | HTTP | | GET | | 18 | Pedido de acesso app JavaScript |
| 17 | PC1 | PC2 | HTTP | 200 OK | | 0.6268 | | Resposta a linha 9 com texto css |
| 18 | PC1 | PC2 | HTTP | 200 OK | | 0.344 | | Resposta a linha 16 com app JavaScript |
| 19 | PC2 | PC1 | HTTP | | GET | | 22 | Pedido de imagem |
| 20 | PC2 | PC1 | HTTP | | GET | | 21 | Pedido de imagem |
| 21 | PC1 | PC2 | HTTP | 200 OK | | 0.0024 | | Resposta a linha 20 PNG |
| 22 | PC1 | PC2 | HTTP | 200 OK | | 0.0025 | | Resposta a linha 19 PNG |
| 23 | PC2 | PC1 | HTTP | | GET | | 24 | Pedido de imagem |
| 24 | PC1 | PC2 | HTTP | 200 OK | | 0.001 | | Resposta a linha 23 PNG |

Figura 7 – Tabela de Packets

As duas primeiras linhas correspondem ao pedido (GET) de acesso ao servidor em localhost e resposta (HTTP 1.1) ao mesmo.

Os restantes packets apresentados são os mesmos mostrados na listagem de packets capturados no PC2 – *Figura 6*.

As duas linhas seguintes representam o pedido (GET) de acesso ao servidor virtual no PC1 por parte do PC2 e a disponibilização (resposta - HTTP 1.1) do mesmo por parte do PC1.

As restantes linhas são como réplicas que completam o pedido original e o chamando todo o conteúdo necessário ao recurso. Em pares de pedidos e respostas.

Template para Headers de Queries HTTP

Hypertext Transfer Protocol

<RequestMethod> <URI> <http version> -> descrição
[Expert Info (Chat/Sequence): <descrição>]
[<descrição>]
[Severity level]
[Group]

Request Method: -> Maneira de lidar com o recurso pedido.

//Neste trabalho é sempre **GET**. Pede uma do representação do recurso.

Request URI: -> Da mesma forma que o URL é um localizador do recurso, o URI é um **identificador** do recurso.

Request Version: -> HTTP version

//**HTTP 1.1** é sempre a versão dos pedidos/respostas.

//0.9 - Toda a info aparecia numa linha - GET <path>. Resposta só //aceita *hypertext*. Não tem *headers*. Ligação termina após resposta.

//1.0 - Organizado por *headers*. Aceita métodos GET, HEAD e POST. //Resposta aceita diferentes tipos de conteúdo. Ligação termina após //resposta.

//1.1 - Melhor desempenho geral através de características como //compressão e descompressão de dados. Aceita mais métodos. Ligação //não fecha após cada resposta - tipo: *long-lived*. Usa de uma ligação //persistente: *keep-alive*.

Host: -> Indica ao servidor qual o *virtual host* a usar(caso haja algum).

Connection: -> Estado/força da ligação

Upgrade-Insecure-Requests: -> Preferências de encriptação/segurança

User-Agent: -> Identifica características do sistema do emissor do pedido.

Accept: -> **Linguagens/Conteúdo Aceites**

Accept-Encoding: -> codificação

Accept-Language: -> Línguas Aceites

[Full request URI: http://<host>/<URI>/]

[**HTTP request x/y**] -> Posição do request na cadeia (ordem de processamento). No campo No. do Wireshark pode -se que alguns packets se encontram em cadeia.

[Prev request in frame:] -> No./frame do request anterior

x - 1 em y

[Response in frame:] -> No./frame do resposta correspondente

i em j

[Next request in frame:] -> No./frame do próximo request

x + 1 em y

Template para Headers de Responses HTTP

Hypertext Transfer Protocol

<Protocol> **<status_code>** **<phrase>** -> descrição
[Expert Info (Chat/Sequence): <descrição>]
[<descrição>]
[Severity level:]
[Group:]

Response Version: -> HTTP version

Status Code: -> Está associado ao header **Response Phrase**. Indica o estado do recurso/reposta.

[Status Code Description: <phrase>]

Response Phrase:

Date: -> data/hora da resposta

Server:

Last-Modified: -> Data da última modificação do recurso acedido

ETag: -> Validação de **cache**

Accept-Ranges: -> **unidade de medida** de dados transmitidos

Content-Length: -> **tamanho** do ficheiro carregado de acordo com a unidade definida

Keep-Alive: -> argumentos para manter a ligação aberta

Connection: -> força/estado da ligação

Content-Type: -> tipo de conteúdo apresentado

[HTTP response i/j] -> (Ver Template de Header de Pedidos)

[Time since request] -> Tempo decorrido desde que é emitido o pedido até que se recebe a resposta.

[Prev request in frame:] -> No./frame do query anterior

x - 1 em y

[Prev response in frame:] -> No./frame da resposta anterior

i - 1 em j

[Request in frame:] -> No./frame do query correspondente

x em y

[Next request in frame:] -> No./frame do proximo query

x + 1 em y

[Next response in frame:] -> No./frame da proxima resposta

i + 1 em j

[Request URI: URI do Query correspondente]

File Data:

Exemplo de Headers de Pedido e da resperiva Resposta HTTP

LINHA 5

```
Hypertext Transfer Protocol
GET /dashboard/ HTTP/1.1\r\n
[Expert Info (Chat/Sequence): GET /dashboard/ HTTP/1.1\r\n]
[GET /dashboard/ HTTP/1.1\r\n]
[Severity level: Chat]
[Group: Sequence]
Request Method: GET
Request URI: /dashboard/
Request Version: HTTP/1.1
Host: 10.10.37.34\r\n
Connection: keep-alive\r\n
Upgrade-Insecure-Requests: 1\r\n
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/73.0.3683.86 Safari/537.36\r\n
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
,application/signed-exchange;v=b3\r\n
Accept-Encoding: gzip, deflate\r\n
Accept-Language: en-GB,en;q=0.9,pt-PT;q=0.8,pt;q=0.7,en-US;q=0.6\r\n
\r\n
[Full request URI: http://10.10.37.34/dashboard/]
[HTTP request 2/4]
[Prev request in frame: 3298]
[Response in frame: 3307]
[Next request in frame: 3310]
```

LINHA 6

```
Hypertext Transfer Protocol
HTTP/1.1 200 OK\r\n
[Expert Info (Chat/Sequence): HTTP/1.1 200 OK\r\n]
[HTTP/1.1 200 OK\r\n]
[Severity level: Chat]
[Group: Sequence]
Response Version: HTTP/1.1
Status Code: 200
[Status Code Description: OK]
Response Phrase: OK
Date: Wed, 27 Mar 2019 19:43:22 GMT\r\n
Server: Apache/2.4.38 (Win64) OpenSSL/1.1.1b PHP/7.3.3\r\n
Last-Modified: Wed, 13 Mar 2019 07:51:17 GMT\r\n
ETag: "1d98-583f51212c740"\r\n
Accept-Ranges: bytes\r\n
Content-Length: 7576\r\n
Keep-Alive: timeout=5, max=99\r\n
Connection: Keep-Alive\r\n
Content-Type: text/html\r\n
\r\n
[HTTP response 2/4]
[Time since request: 0.003259000 seconds]
[Prev request in frame: 3298]
[Prev response in frame: 3300]
[Request in frame: 3301]
[Next request in frame: 3310]
[Next response in frame: 3315]
[Request URI: http://10.10.37.34/dashboard/]
File Data: 7576 bytes
```

Conclusão

Acesso ao *webserver* quando ambos os computadores se encontram na mesma rede é bastante direto, sabendo o endereço IP do *host*.

Através de rede externa mostrou-se mais trabalhoso visto requerer modificações no *router* da rede do *host*.

Solução encontrada foi o uso de ferramenta de *Tunneling*. No entanto não foi testado.

Avaliação dos *packets* capturados permitiu compreender a complexidade por trás de uma simples procura de uma página *web*.