TP3-SCR

Diogo Aires, A91685
João Silva, A91638
Eduardo Pereira, A70619

2. Nível Aplicacional

529	73.271044	172.26.13.120	193.137.16.65	DNS	77 Standard query 0xbb79 A www.sas.uminho.pt
530	73.274298	193.137.16.65	172.26.13.120	DNS	281 Standard query response 0xbb79 A www.sas.uminho.pt

Pergunta 1) ***Identifique o endereço IP da estação que formulou a query DNS e o tipo de query realizada. (Nota: Caso não consiga encontrar a referida query, limpe a cache DNS da sua máquina, executando num terminal do Ubuntu: sudo systemd-resolve --flush-caches; ou sudo /etc/init.d/dns-clean restart. No Windows deve executar o comando: ipconfig /flushdns).***

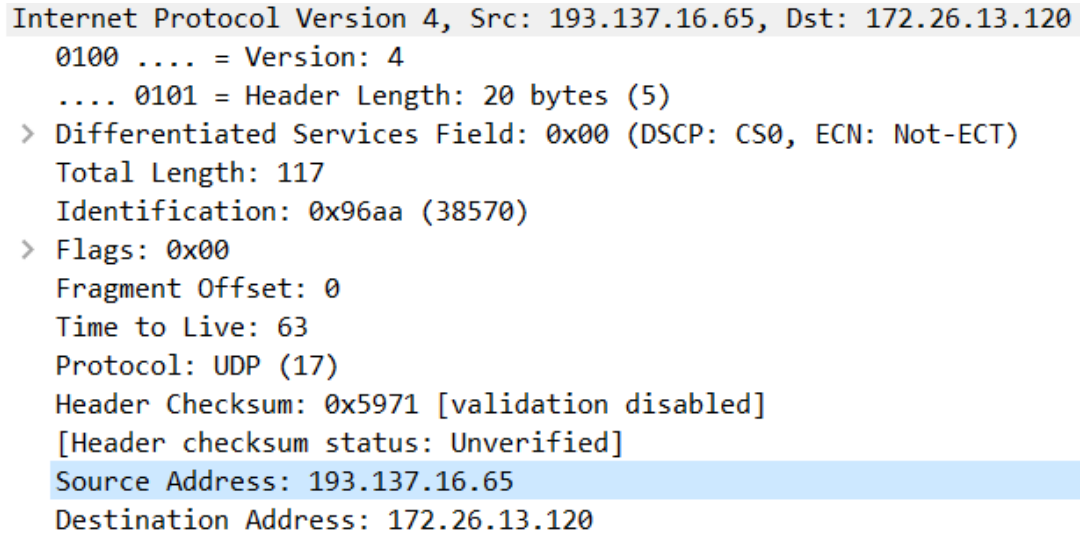
```
Internet Protocol Version 4, Src: 172.26.13.120, Dst: 193.137.16.65
   0100 .... = Version: 4
   .... 0101 = Header Length: 20 bytes (5)
 > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
   Total Length: 63
   Identification: 0x0c19 (3097)
 > Flags: 0x00
   Fragment Offset: 0
   Time to Live: 128
   Protocol: UDP (17)
   Header Checksum: 0x0000 [validation disabled]
   [Header checksum status: Unverified]
   Source Address: 172.26.13.120
   Destination Address: 193.137.16.65
```

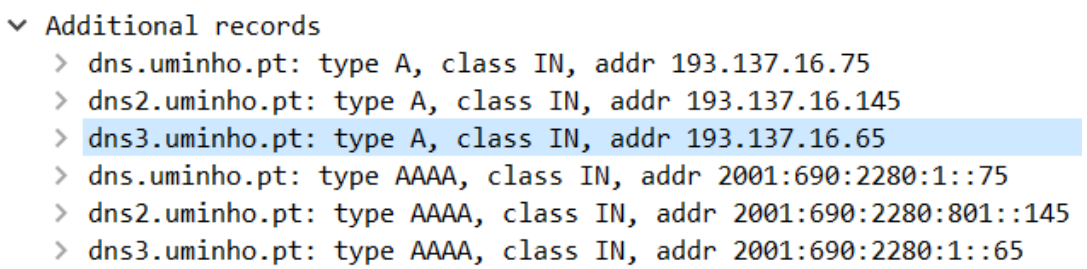
O endereço IP da estação que formulou a query DNS, neste caso a minha máquina, é 172.26.13.120 .

```
Domain Name System (query)
   Transaction ID: 0xbb79
 > Flags: 0x0100 Standard query
   Questions: 1
   Answer RRs: 0
   Authority RRs: 0
   Additional RRs: 0
 v Queries
   > www.sas.uminho.pt: type A, class IN
   [Response In: 530]
```

A query formulada é de tipo A, que é utilizada quando pretendemos saber o IPv4.

Pergunta 2) ***Localize a trama com a resposta à query DNS formulada. Identifique nesta trama o endereço IP do servidor web. Identifique também o servidor de nomes que forneceu a resposta, através do seu IP e nome (sugestão: consulte o Additional Records).***





O endereço IP do servidor é 193.137.16.65 . Como podemos ver nas imagens acima, este endereço IP corresponde ao IP do DNS3.

HTTP e TCP

Pergunta 3) Aplique o filtro aos protocolos http // tcp. Identifique os endereços IP do cliente e do servidor HTTP.

78	3.100791	172.26.13.120	193.137.9.178	TCP	66 59785 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
79	3.101198	172.26.13.120	193.137.9.178	TCP	66 59786 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
80	3.102783	193.137.9.178	172.26.13.120	TCP	66 80 → 59785 [SYN, ACK] Seq=0 Ack=1 Win=16384 Len=0 MSS=1250 WS=1 SACK_PERM=1
81	3.102997	172.26.13.120	193.137.9.178	TCP	54 59785 → 80 [ACK] Seq=1 Ack=1 Win=131072 Len=0
82	3.103450	172.26.13.120	193.137.9.178	HTTP	547 GET / HTTP/1.1
83	3.104186	193.137.9.178	172.26.13.120	TCP	66 80 → 59786 [SYN, ACK] Seq=0 Ack=1 Win=16384 Len=0 MSS=1250 WS=1 SACK_PERM=1
84	3.104264	172.26.13.120	193.137.9.178	TCP	54 59786 → 80 [ACK] Seq=1 Ack=1 Win=131072 Len=0

O endereço IP do cliente continua a ser o da minha máquina, ou seja, é o endereço 172.26.13.120 e o endereço do servidor HTTP é 192.137.9.178 .

Pergunta 4) Identifique os segmentos TCP correspondentes ao estabelecimento da ligação entre o cliente e o servidor HTTP. Qual o tamanho máximo de segmento (MSS) que o servidor aceita receber?

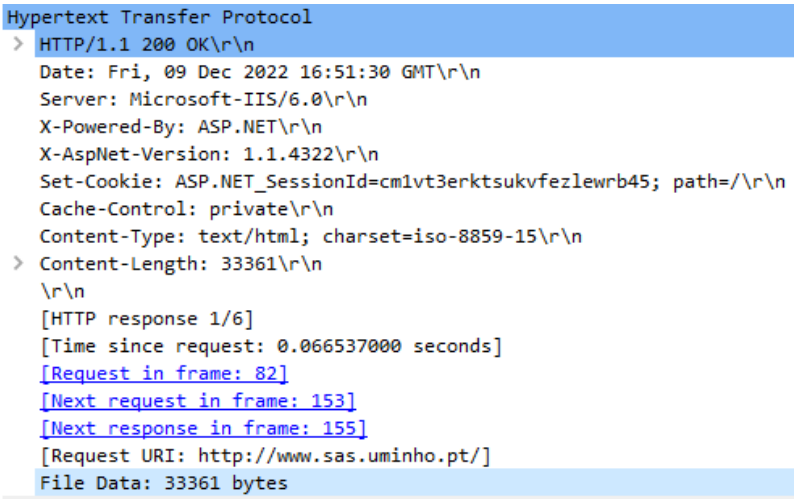
78	3.100791	172.26.13.120	193.137.9.178	TCP	66 59785 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
79	3.101198	172.26.13.120	193.137.9.178	TCP	66 59786 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
80	3.102783	193.137.9.178	172.26.13.120	TCP	66 80 → 59785 [SYN, ACK] Seq=0 Ack=1 Win=16384 Len=0 MSS=1250 WS=1 SACK_PERM=1
81	3.102997	172.26.13.120	193.137.9.178	TCP	54 59785 → 80 [ACK] Seq=1 Ack=1 Win=131072 Len=0

Como podemos ver na imagem acima, na linha 80, o servidor indica ao cliente que o MSS que aceita receber é 1520 bytes.

86	3.158007	193.137.9.178	172.26.13.120	TCP	1304 80 → 59785 [ACK] Seq=1251 Ack=494 Win=65042 Len=1250 [TCP segment of a reassembled PDU]
87	3.158046	172.26.13.120	193.137.9.178	TCP	54 59785 → 80 [ACK] Seq=494 Ack=2501 Win=131072 Len=0
88	3.160869	193.137.9.178	172.26.13.120	TCP	1304 80 → 59785 [ACK] Seq=2501 Ack=494 Win=65042 Len=1250 [TCP segment of a reassembled PDU]
89	3.160869	193.137.9.178	172.26.13.120	TCP	1304 80 → 59785 [ACK] Seq=3751 Ack=494 Win=65042 Len=1250 [TCP segment of a reassembled PDU]
90	3.160869	193.137.9.178	172.26.13.120	TCP	1304 80 → 59785 [ACK] Seq=5001 Ack=494 Win=65042 Len=1250 [TCP segment of a reassembled PDU]

No campo length, pode-se ver que o tamanho não ultrapassa 1520 bytes.

Pergunta 5) ***Identifique a resposta HTTP do servidor respeitante ao primeiro pedido GET efetuado pelo cliente. Quantos bytes de dados aplicacionais contém essa resposta HTTP?***



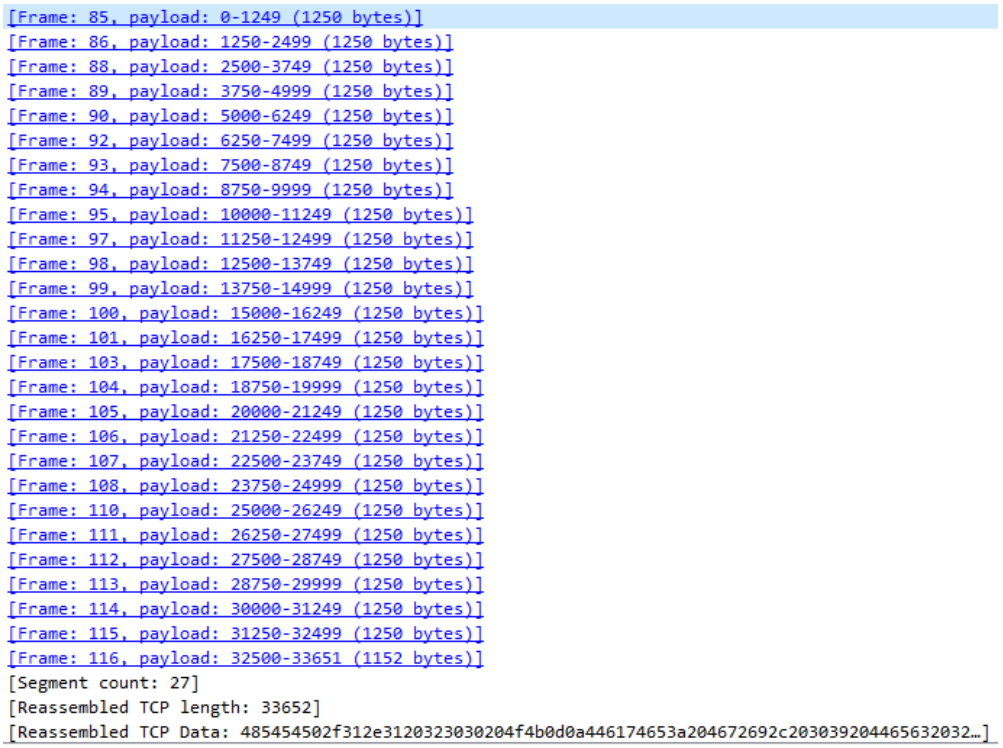
De acordo com a imagem acima, na resposta HTTP ao pedido GET feito pelo cliente, podemos ver que esta contém 33361 bytes de dados.

Pergunta 6) ***A resposta HTTP identificada na alínea anterior foi transmitida em quantos segmentos TCP? Apresente também uma estimativa teórica para essa quantidade.***

Teoricamente, a resposta HTTP deve ser transmitida em 27 segmentos. Pois tem de conter os 33361 bytes de dados, porém, como o MSS é apenas 1250 bytes, estes 33361 bytes têm de ser divididos. Ou seja, se dividirmos 33361 por 1250, podemos ver que **33361/1250=26,6888**. Então a resposta HTTP precisa de ser transmitida em 26 segmentos com 1250 bytes e é necessário mais um segmento para transmitir os bytes que restam.

Analisando a imagem acima, conseguimos confirmar que é exatamente esse o número de segmentos em que a resposta HTTP é transmitida.

Pergunta 7) A partir da informação contida nos cabeçalhos dos protocolos IP e TCP, determine o número de bytes de dados enviados no primeiro e no último segmento TCP respeitantes à resposta HTTP.

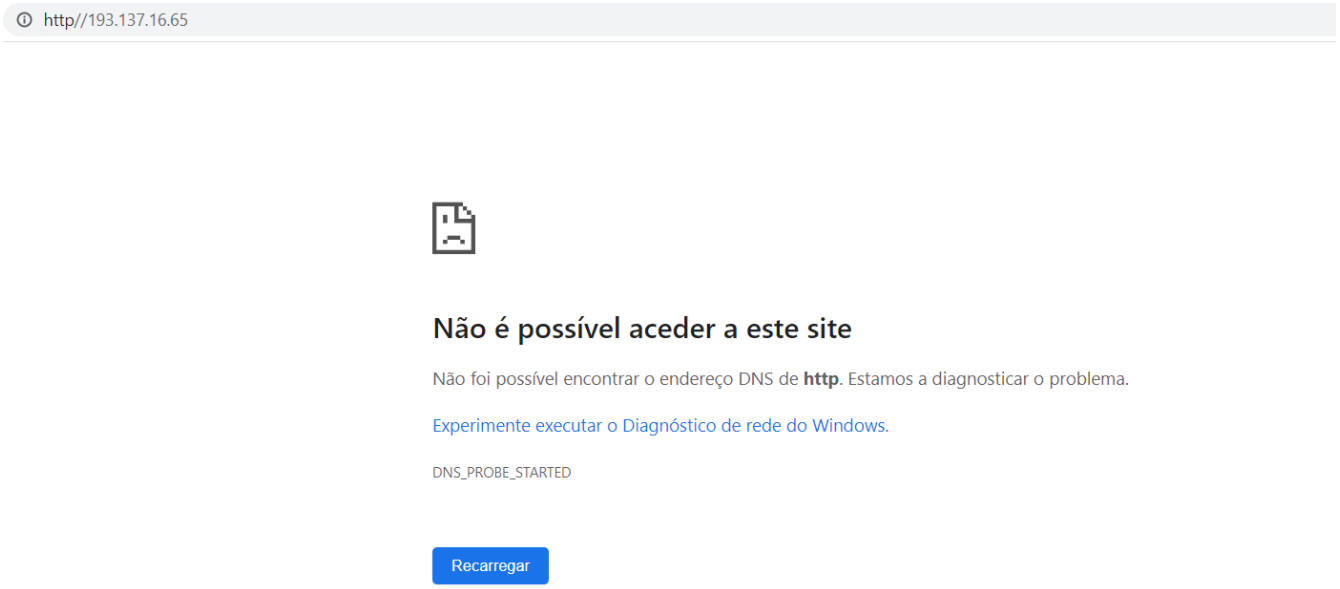


O número de bytes enviados no primeiro segmento TCP da resposta HTTP foi de 1250 bytes, devido ao MSS ser 1250 bytes. No caso do último segmento TCP, este contém apenas 1152 bytes de dados, pois, como vimos na **Pergunta 6)**, 26 segmentos não eram suficientes para a transmissão da resposta HTTP, pelo que este último segmento transporta os restantes bytes.

Pergunta 8) ***Observe a informação apresentada no campo host do cabeçalho do pedido HTTP e diga qual o seu interesse? Experimente aceder à mesma página web através de http://endereço_IP, em que endereço_IP é o respeitante a www.sas.uminho.pt (identificado na alínea 2). Justifique o comportamento observado.***

```
Hypertext Transfer Protocol
> GET / HTTP/1.1\r\n
  Host: www.sas.uminho.pt\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/108.0.0.0 Safari/537.36\r\n
  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9\r\n
  Accept-Encoding: gzip, deflate\r\n
  Accept-Language: pt-PT,pt;q=0.9,en-US;q=0.8,en;q=0.7\r\n
```

A informação apresentada no campo host serve para saber a que página web queremos aceder, neste caso www.sas.uminho.pt.



Como podemos ver ao tentar aceder à página web através de http://193.137.16.65, não obtemos nenhuma resposta, dá erro. Isto acontece porque, em cada servidor, pode existir mais que uma página web, ou seja, se tentarmos aceder só pelo IP correspondente ao servidor, não obtemos resposta. Como este é o caso do servidor respeitante a www.sas.uminho.pt, a procura através do IP do servidor (193.137.16.65), não obtém resposta.

Caso o servidor apenas suportasse uma única página web, esta pesquisa já não daria erro, pois o IP do servidor estaria apenas associado a essa página web.

Pergunta 9) ***Com base na sequência de dados trocados entre o cliente e o servidor diga, justificando, se o servidor HTTP está a funcionar em modo de conexão persistente ou não persistente.***

Diz-se que uma conexão é não persistente quando, após realizar a transferência de um objeto entre cliente e servidor, esta conexão TCP é encerrada, ou seja, não funciona para outros objetos da página web.

No caso da conexão ser persistente, a conexão fica disponível para outros pedidos entre o cliente e o servidor. Então, como se pode verificar, esta conexão é persistente pois não é terminada após a transmissão do primeiro objeto.

Pergunta 10) ***Aplique o filtro apenas ao protocolo http. O hard refresh permite limpar a cache do browser para uma determinada página, forçando o browser a carregar a última versão da página existente no servidor. Normalmente o hard refresh duma página faz-se com CTRL+F5 ou SHIFT+page reload (caso não funcione, procure na Internet a forma de fazer hard refresh no seu browser). Coloque o Wireshark a capturar tráfego e faça hard refresh da página indicada anteriormente. Depois volte a aceder à mesma página mas sem fazer hard refresh. Pare a captura de tráfego. Identifique a principal diferença observada no tráfego HTTP entre carregar a referida página com e sem hard refresh. Qual a principal vantagem e desvantagem inerente ao hard refresh?***

A seguinte imagem mostra a captura do tráfego feita pelo Wireshark quando se faz Hard Refresh.

	Time	Source	Destination	Protocol	Length	Info
13	0.421517	172.26.13.120	193.137.9.178	HTTP	634	GET / HTTP/1.1
43	0.519368	193.137.9.178	172.26.13.120	HTTP	106	HTTP/1.1 200 OK (text/html)
45	0.522290	172.26.13.120	193.137.9.178	HTTP	533	GET /portal.css HTTP/1.1
47	0.522651	172.26.13.120	193.137.9.178	HTTP	526	GET /lib/clientUtils.js HTTP/1.1
50	0.524858	172.26.13.120	193.137.9.178	HTTP	527	GET /lib/1k_standards.js HTTP/1.1
54	0.534986	193.137.9.178	172.26.13.120	HTTP	87	HTTP/1.1 200 OK (application/x-javascript)
57	0.536813	193.137.9.178	172.26.13.120	HTTP	1222	HTTP/1.1 200 OK (application/x-javascript)
58	0.538767	172.26.13.120	193.137.9.178	HTTP	599	GET /images/escolas/corReitoria.gif HTTP/1.1
59	0.550422	193.137.9.178	172.26.13.120	HTTP	1206	HTTP/1.1 200 OK (GIF89a)
60	0.551765	172.26.13.120	193.137.9.178	HTTP	593	GET /images/globais/en-us.gif HTTP/1.1
62	0.560290	193.137.9.178	172.26.13.120	HTTP	245	HTTP/1.1 200 OK (GIF89a)
64	0.561548	172.26.13.120	193.137.9.178	HTTP	592	GET /images/tab/um_pt-PT.gif HTTP/1.1
66	0.568977	193.137.9.178	172.26.13.120	HTTP	1261	HTTP/1.1 200 OK (GIF89a)
68	0.570105	172.26.13.120	193.137.9.178	HTTP	592	GET /images/tab/servicos.gif HTTP/1.1
69	0.576697	193.137.9.178	172.26.13.120	HTTP	944	HTTP/1.1 200 OK (GIF89a)
70	0.578002	172.26.13.120	193.137.9.178	HTTP	595	GET /images/tab/Alimentacao.gif HTTP/1.1
71	0.584525	193.137.9.178	172.26.13.120	HTTP	733	HTTP/1.1 200 OK (GIF89a)
72	0.585768	172.26.13.120	193.137.9.178	HTTP	594	GET /images/tab/Alojamento.gif HTTP/1.1
73	0.592235	193.137.9.178	172.26.13.120	HTTP	705	HTTP/1.1 200 OK (GIF89a)
74	0.593401	172.26.13.120	193.137.9.178	HTTP	589	GET /images/tab/apoio.gif HTTP/1.1
75	0.598352	193.137.9.178	172.26.13.120	HTTP	678	HTTP/1.1 200 OK (GIF89a)
76	0.599557	172.26.13.120	193.137.9.178	HTTP	590	GET /images/tab/Bolsas.gif HTTP/1.1
93	0.602571	193.137.9.178	172.26.13.120	HTTP	1162	HTTP/1.1 200 OK (text/css)
95	0.603410	172.26.13.120	193.137.9.178	HTTP	592	GET /images/tab/Desporto.gif HTTP/1.1
96	0.604517	172.26.13.120	193.137.9.178	HTTP	589	GET /images/tab/Links.gif HTTP/1.1
97	0.604577	193.137.9.178	172.26.13.120	HTTP	608	HTTP/1.1 200 OK (GIF89a)
103	0.607274	172.26.13.120	193.137.9.178	HTTP	593	GET /images/globais/login.gif HTTP/1.1
104	0.607337	172.26.13.120	193.137.9.178	HTTP	592	GET /images/globais/home.gif HTTP/1.1
105	0.608336	193.137.9.178	172.26.13.120	HTTP	899	HTTP/1.1 200 OK (GIF89a)
106	0.608853	172.26.13.120	193.137.9.178	HTTP	592	GET /images/globais/Mapa.gif HTTP/1.1
111	0.609087	172.26.13.120	193.137.9.178	HTTP	597	GET /images/globais/Contactos.gif HTTP/1.1
112	0.609174	172.26.13.120	193.137.9.178	HTTP	597	GET /images/globais/email_web.gif HTTP/1.1
113	0.610028	193.137.9.178	172.26.13.120	HTTP	496	HTTP/1.1 200 OK (GIF89a)
114	0.610479	172.26.13.120	193.137.9.178	HTTP	593	GET /images/globais/print.gif HTTP/1.1
115	0.612017	193.137.9.178	172.26.13.120	HTTP	692	HTTP/1.1 200 OK (GIF89a)
116	0.612715	172.26.13.120	193.137.9.178	HTTP	586	GET /images/spacer.gif HTTP/1.1
117	0.613610	193.137.9.178	172.26.13.120	HTTP	929	HTTP/1.1 200 OK (GIF89a)

A imagem abaixo mostra o tráfego capturado pelo Wireshark, desta vez apenas fazendo Refresh.

	Time	Source	Destination	Protocol	Length	Info
30	3.527425	172.26.13.120	193.137.9.178	HTTP	617	GET / HTTP/1.1
60	3.621836	193.137.9.178	172.26.13.120	HTTP	106	HTTP/1.1 200 OK (text/html)

A principal diferença é que o Hard Refresh tem de ir buscar todas as informações ao servidor, sendo que se houver alguma atualização realizada pelo servidor, um simples Refresh não capta a informação toda.

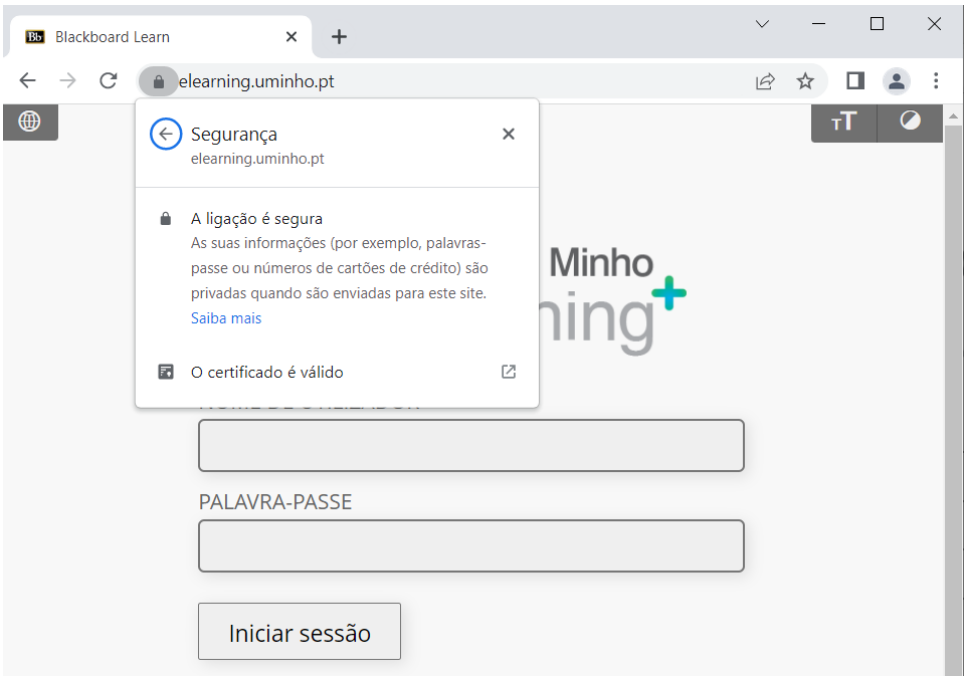
O Hard Refresh tem a vantagem de, caso ocorra alguma atualização por parte do servidor, capta a informação toda, ao contrário do Refresh que, neste caso, perde informação e não permite que o utilizador receba a informação mais atualizada.

Se analisarmos a eficiência do Hard Refresh face ao Refresh, vemos que esta é a sua desvantagem, pois enquanto o Refresh não precisa de consultar o servidor para a informação toda, o Hard Refresh, mesmo que não haja nenhuma atualização, pede ao servidor todas as informações de novo, o que custa tempo e, possivelmente, sobrecarrega mais a rede, logo, é menos eficiente.

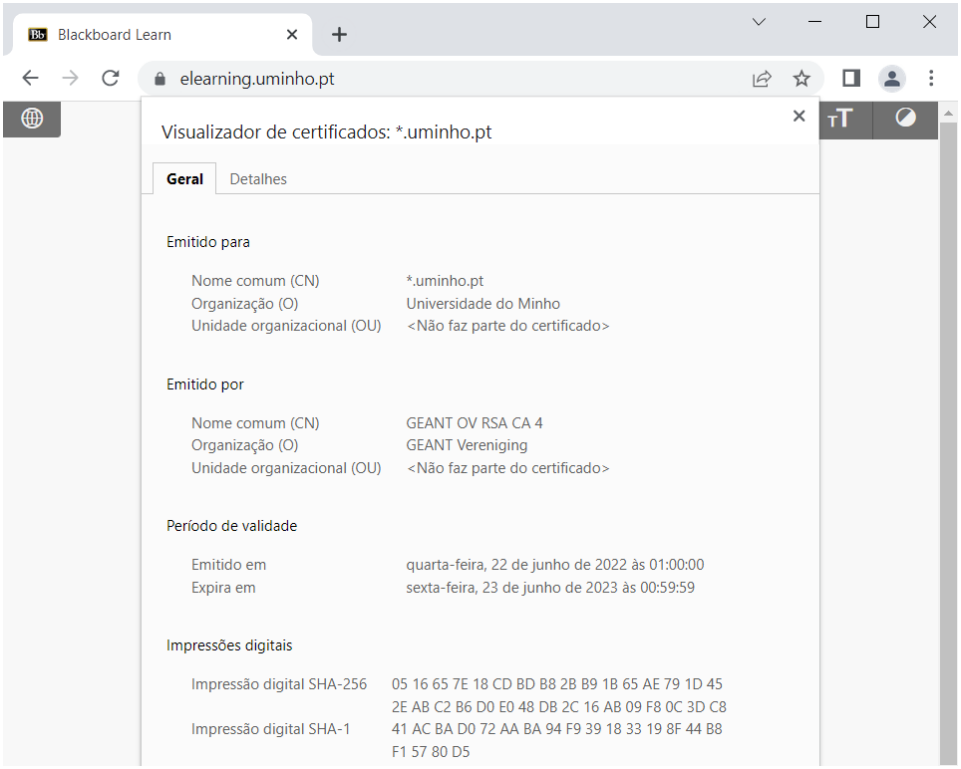
HTTPS

Pergunta 11) Aceda a https://elearning.uminho.pt, ao mesmo tempo que captura o tráfego desse acesso com o Wireshark.

(a) De que forma o seu browser assinala que o utilizador está perante, ou não, uma ligação HTTP ao servidor segura? Apresente uma captura de écran com essa indicação.



Quando o browser numa determinada página web tem um ícone de um aloquete no canto superior esquerdo, significa que a ligação HTTP é segura.



Pelas informações descritas na imagem acima, nomeadamente a data em que foi emitido e a data em que expira, confirma-se que o certificado que comprova que esta ligação é segura, está válido.

(b) Porque razão o tráfego HTTP não é identificado como tal no Wireshark? Apesar disso, pode detetar-se qual o protocolo aplicacional. Como é que o Wireshark sabe que se trata duma ligação http-over-tls?

O Wireshark não consegue decifrar, por se tratar de uma ligação segura, por isso não lê como HTTP, mas sim como TLS.

Pergunta 12) Diga, justificando, quais dos seguintes elementos uma comunicação HTTPS permite manter ocultos dum atacante:

i) o endereço IP do cliente,
ii) o endereço IP do servidor web
iii) o nome do servidor web
iv) o tamanho da mensagem trocada entre o cliente o servidor,
v) a identificação da página acedida no servidor web,
vi) a frequência das conexões estabelecidas entre o cliente e o servidor,
vii) os dados da aplicação trocados entre o servidor e o cliente.

Time	Source	Destination	Protocol	Length	Info
0.000000	172.26.13.120	142.250.200.110	UDP	76	49566 → 443 Len=34
0.022897	142.250.200.110	172.26.13.120	UDP	68	443 → 49566 Len=26
1.849981	172.26.13.120	193.137.16.65	DNS	79	Standard query 0x7d12 A elearning.uminho.pt
1.850122	172.26.13.120	193.137.16.65	DNS	79	Standard query 0x19d2 HTTPS elearning.uminho.pt
1.852074	193.137.16.65	172.26.13.120	DNS	351	Standard query response 0x7d12 A elearning.uminho.pt A 193.137.9.150 NS dns2.uminho.pt
1.852074	193.137.16.65	172.26.13.120	DNS	133	Standard query response 0x19d2 HTTPS elearning.uminho.pt SOA dns.uminho.pt
1.852442	172.26.13.120	193.137.9.150	TCP	66	62295 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
1.852669	172.26.13.120	193.137.9.150	TCP	66	62296 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
1.854121	193.137.9.150	172.26.13.120	TCP	66	443 → 62295 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1250 SACK_PERM=1 WS=64
1.854156	172.26.13.120	193.137.9.150	TCP	54	62295 → 443 [ACK] Seq=1 Ack=1 Win=131072 Len=0
1.854270	172.26.13.120	193.137.9.150	TLSv1.2	571	Client Hello
1.855042	193.137.9.150	172.26.13.120	TCP	66	443 → 62296 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1250 SACK_PERM=1 WS=64
1.855061	172.26.13.120	193.137.9.150	TCP	54	62296 → 443 [ACK] Seq=1 Ack=1 Win=131072 Len=0
1.855144	172.26.13.120	193.137.9.150	TLSv1.2	571	Client Hello

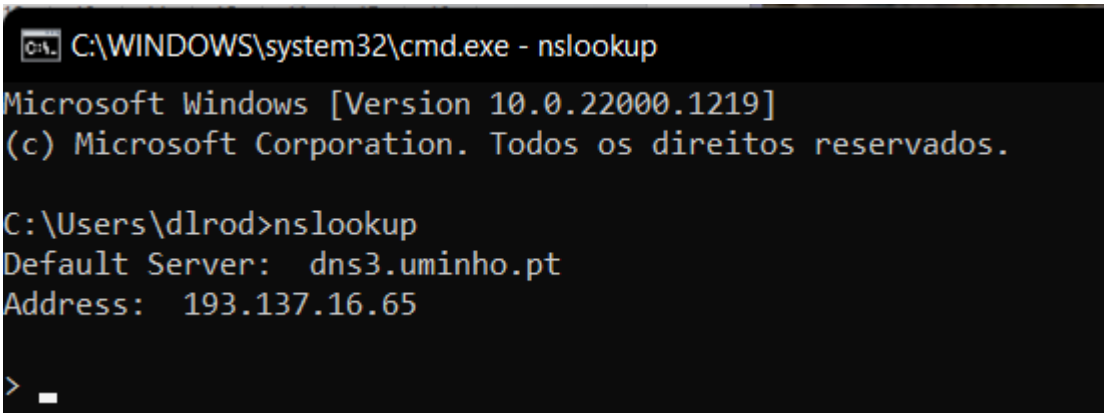
Ao analisar o tráfego no Wireshark, podemos ver que os únicos elementos que a comunicação HTTP permite ocultar de um atacante é:

v) identificação da página
vii) os dados da aplicação trocados entre servidor e cliente (por exemplo, passwords, etc).

Todos os outros elementos são visíveis através da captura do tráfego.

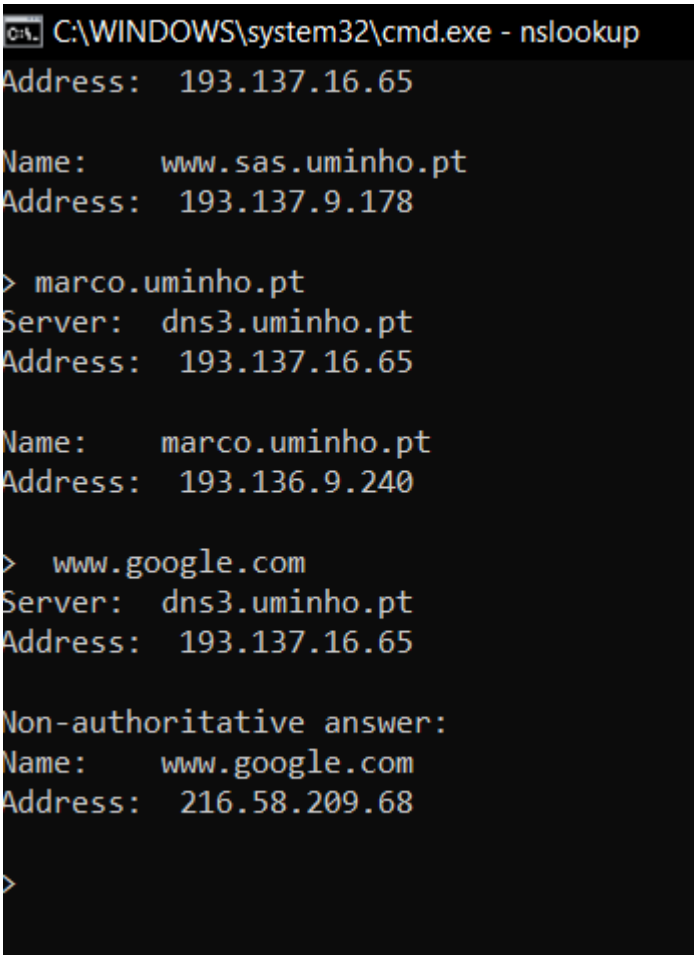
3. Consultas ao serviço de resolução de nomes DNS

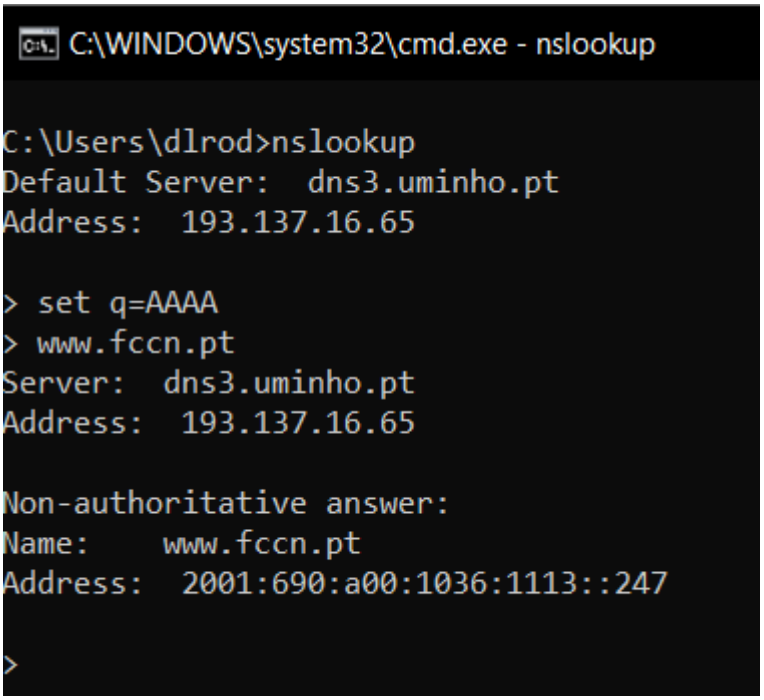
Pergunta 1) ***Se estiver a usar o Linux, observe o conteúdo do ficheiro /etc/resolv.conf. Se estiver a usar o Windows, abra uma janela de comandos e execute nslookup. Indique qual o servidor de nomes que a sua máquina está a usar?***



Conseguimos observar que o servidor que a máquina está a usar é: dns3.uminho.pt

Pergunta 2) ***Usando o registo do tipo A, identifique os endereços IPv4 dos servidores www.sas.uminho.pt, marco.uminho.pt e www.google.com? Usando o registo AAAA, identifique o endereço IPv6 do servidor www.fccn.pt.***





IPv4:
- www.sas.uminho.pt -> 193.137.9.178
- marco.uminho.pt -> 193.137.9.240
- www.google.com -> 216.58.209.68

IPv6:
- www.fccn.pt -> 2001:690:a00:1036:1113::247

Pergunta 3) ***Experimente fazer uma query aos registos PTR para os nomes 240.9.136.193.in-addr.arpa. e 7.4.2.0.0.0.0.0.0.0.0.0.0.3.1.1.1.6.3.0.1.0.0.a.0.0.9.6.0.1.0.0.2.ip6.arpa. Comente os resultados face aos obtidos na alínea anterior.***

```
> set q=PTR
> 240.9.136.193.in-addr.arpa.
Server:  dns3.uminho.pt
Address:  193.137.16.65

240.9.136.193.in-addr.arpa      name = marco.uminho.pt
9.136.193.in-addr.arpa  nameserver = dns2.uminho.pt
9.136.193.in-addr.arpa  nameserver = dns.uminho.pt
9.136.193.in-addr.arpa  nameserver = ns-rev.dns.pt
9.136.193.in-addr.arpa  nameserver = animal.inescn.pt
9.136.193.in-addr.arpa  nameserver = dns3.uminho.pt
9.136.193.in-addr.arpa  nameserver = marco.uminho.pt
marco.uminho.pt internet address = 193.136.9.240
dns.uminho.pt   internet address = 193.137.16.75
dns2.uminho.pt  internet address = 193.137.16.145
dns3.uminho.pt  internet address = 193.137.16.65
dns.uminho.pt   AAAA IPv6 address = 2001:690:2280:1::75
dns2.uminho.pt  AAAA IPv6 address = 2001:690:2280:801::145
dns3.uminho.pt  AAAA IPv6 address = 2001:690:2280:1::65
> 7.4.2.0.0.0.0.0.0.0.0.0.0.3.1.1.1.6.3.0.1.0.0.a.0.0.9.6.0.1.0.0.2.ip6.arpa.
Server:  dns3.uminho.pt
Address:  193.137.16.65

Non-authoritative answer:
7.4.2.0.0.0.0.0.0.0.0.0.0.3.1.1.1.6.3.0.1.0.0.a.0.0.9.6.0.1.0.0.2.ip6.arpa       name = www.fccn.pt

0.9.6.0.1.0.0.2.ip6.arpa        nameserver = ns03.fccn.pt
0.9.6.0.1.0.0.2.ip6.arpa        nameserver = ns01.fccn.pt
0.9.6.0.1.0.0.2.ip6.arpa        nameserver = ns02.fccn.pt
ns01.fccn.pt    internet address = 193.136.192.40
ns02.fccn.pt    internet address = 193.136.2.228
ns03.fccn.pt    internet address = 138.246.255.249
ns01.fccn.pt    AAAA IPv6 address = 2001:690:a00:4001::200
ns02.fccn.pt    AAAA IPv6 address = 2001:690:a80:4001::200
ns03.fccn.pt    AAAA IPv6 address = 2001:4ca0:106:0:250:56ff:fea9:3fd
>
```

O DNS permite conhecer o nome a partir do endereço IP, neste caso a partir do endereço IPv4 “193.136.9.240” reconheceu o “marco.uminho.pt” e a partir do endereço IPv6 “2001:690:a00:1036:1113::247” reconheceu “www.fccn.pt”.

Pergunta 4) ***Usando os registo NS, identifique os servidores de nomes definidos para os domínios: “uminho.com.”, “sas.uminho.pt.”, “pt.” e “.” (root).***

i) Perante a informação obtida, diga, justificando, se os servidores de nomes de diferentes domínios podem coexistir numa mesma máquina física.

ii) Encontra domínios geridos por servidores de nomes localizados em redes IP distintas? Se sim, apresente esses domínios e diga qual a vantagem resultante desse procedimento?

```
C:\WINDOWS\system32\cmd.exe - nslookup

C:\Users\dlrod>nslookup
Default Server:  dns3.uminho.pt
Address:  193.137.16.65

> set q=NS
> uminho.com.
Server:  dns3.uminho.pt
Address:  193.137.16.65

*** dns3.uminho.pt can't find uminho.com.: Non-existent domain
> uminho.pt.
Server:  dns3.uminho.pt
Address:  193.137.16.65

uminho.pt       nameserver = dns2.uminho.pt
uminho.pt       nameserver = ns02.fccn.pt
uminho.pt       nameserver = dns3.uminho.pt
uminho.pt       nameserver = dns.uminho.pt
dns.uminho.pt   internet address = 193.137.16.75
dns2.uminho.pt  internet address = 193.137.16.145
dns3.uminho.pt  internet address = 193.137.16.65
ns02.fccn.pt    internet address = 193.136.2.228
dns.uminho.pt   AAAA IPv6 address = 2001:690:2280:1::75
dns2.uminho.pt  AAAA IPv6 address = 2001:690:2280:801::145
dns3.uminho.pt  AAAA IPv6 address = 2001:690:2280:1::65
ns02.fccn.pt    AAAA IPv6 address = 2001:690:a80:4001::200
> sas.uminho.pt.
Server:  dns3.uminho.pt
Address:  193.137.16.65

sas.uminho.pt   nameserver = dns2.uminho.pt
sas.uminho.pt   nameserver = dns.uminho.pt
sas.uminho.pt   nameserver = dns3.uminho.pt
dns.uminho.pt   internet address = 193.137.16.75
dns2.uminho.pt  internet address = 193.137.16.145
dns3.uminho.pt  internet address = 193.137.16.65
dns.uminho.pt   AAAA IPv6 address = 2001:690:2280:1::75
dns2.uminho.pt  AAAA IPv6 address = 2001:690:2280:801::145
dns3.uminho.pt  AAAA IPv6 address = 2001:690:2280:1::65
```

```
> pt.
Server:  dns3.uminho.pt
Address:  193.137.16.65

Non-authoritative answer:
pt      nameserver = ns.dns.br
pt      nameserver = b.dns.pt
pt      nameserver = h.dns.pt
pt      nameserver = ns2.nic.fr
pt      nameserver = d.dns.pt
pt      nameserver = e.dns.pt
pt      nameserver = a.dns.pt
pt      nameserver = c.dns.pt
pt      nameserver = g.dns.pt

c.dns.pt        internet address = 204.61.216.105
h.dns.pt        internet address = 194.146.106.138
d.dns.pt        internet address = 185.39.210.1
a.dns.pt        internet address = 185.39.208.1
b.dns.pt        internet address = 194.0.25.23
e.dns.pt        internet address = 193.136.192.64
g.dns.pt        internet address = 193.136.2.226
ns.dns.br       internet address = 200.160.0.5
ns2.nic.fr      internet address = 192.93.0.4
c.dns.pt        AAAA IPv6 address = 2001:500:14:6105:ad::1
h.dns.pt        AAAA IPv6 address = 2001:67c:1010:35::53
d.dns.pt        AAAA IPv6 address = 2a04:6d82::1
a.dns.pt        AAAA IPv6 address = 2a04:6d80::1
b.dns.pt        AAAA IPv6 address = 2001:678:20::23
e.dns.pt        AAAA IPv6 address = 2001:690:a00:4001::64
```

```
> .
Server:  dns3.uminho.pt
Address:  193.137.16.65

Non-authoritative answer:
(root)  nameserver = g.root-servers.net
(root)  nameserver = i.root-servers.net
(root)  nameserver = e.root-servers.net
(root)  nameserver = h.root-servers.net
(root)  nameserver = l.root-servers.net
(root)  nameserver = c.root-servers.net
(root)  nameserver = f.root-servers.net
(root)  nameserver = a.root-servers.net
(root)  nameserver = m.root-servers.net
(root)  nameserver = j.root-servers.net
(root)  nameserver = b.root-servers.net
(root)  nameserver = d.root-servers.net
(root)  nameserver = k.root-servers.net

a.root-servers.net      internet address = 198.41.0.4
b.root-servers.net      internet address = 199.9.14.201
c.root-servers.net      internet address = 192.33.4.12
d.root-servers.net      internet address = 199.7.91.13
e.root-servers.net      internet address = 192.203.230.10
f.root-servers.net      internet address = 192.5.5.241
g.root-servers.net      internet address = 192.112.36.4
h.root-servers.net      internet address = 198.97.190.53
i.root-servers.net      internet address = 192.36.148.17
j.root-servers.net      internet address = 192.58.128.30
k.root-servers.net      internet address = 193.0.14.129
l.root-servers.net      internet address = 199.7.83.42
m.root-servers.net      internet address = 202.12.27.33
a.root-servers.net      AAAA IPv6 address = 2001:503:ba3e::2:30
b.root-servers.net      AAAA IPv6 address = 2001:500:200::b
>
```

i) “Os servidores de nomes de diferentes domínios podem coexistir numa mesma máquina física.”. A afirmação é verdadeira; de facto conseguimos observar por exemplo para os domínios “uminho.pt.”, “sas.uminho.pt.” o servidor “dns3.uminho.pt” coexiste para ambos, na mesma máquina física.

ii) Há domínios geridos por servidores de nomes localizados em redes IP distintas? Sim.

```
> uminho.pt.
Server:  dns3.uminho.pt
Address:  193.137.16.65

uminho.pt       nameserver = dns2.uminho.pt
uminho.pt       nameserver = ns02.fccn.pt
uminho.pt       nameserver = dns3.uminho.pt
uminho.pt       nameserver = dns.uminho.pt
dns.uminho.pt   internet address = 193.137.16.75
dns2.uminho.pt  internet address = 193.137.16.145
dns3.uminho.pt  internet address = 193.137.16.65
ns02.fccn.pt    internet address = 193.136.2.228
dns.uminho.pt   AAAA IPv6 address = 2001:690:2280:1::75
dns2.uminho.pt  AAAA IPv6 address = 2001:690:2280:801::145
dns3.uminho.pt  AAAA IPv6 address = 2001:690:2280:1::65
ns02.fccn.pt    AAAA IPv6 address = 2001:690:a80:4001::200
```

Pergunta 5) ***Usando o registo SOA, identifique o servidor DNS primário definido para os domínios "uminho.pt.", "pt." e "." ? Em que difere o servidor primário de um servidor secundário e qual o significado dos parâmetros temporais associados ao servidor primário?***

```
> set q=SOA
> uminho.pt.
Server:  dns3.uminho.pt
Address:  193.137.16.65

uminho.pt
        primary name server = dns.uminho.pt
        responsible mail addr = servicos.scom.uminho.pt
        serial  = 2022120616
        refresh = 14400 (4 hours)
        retry   = 7200 (2 hours)
        expire  = 1209600 (14 days)
        default TTL = 300 (5 mins)
uminho.pt       nameserver = dns2.uminho.pt
uminho.pt       nameserver = dns3.uminho.pt
uminho.pt       nameserver = dns.uminho.pt
uminho.pt       nameserver = ns02.fccn.pt
dns.uminho.pt   internet address = 193.137.16.75
dns2.uminho.pt  internet address = 193.137.16.145
dns3.uminho.pt  internet address = 193.137.16.65
ns02.fccn.pt    internet address = 193.136.2.228
dns.uminho.pt   AAAA IPv6 address = 2001:690:2280:1::75
dns2.uminho.pt  AAAA IPv6 address = 2001:690:2280:801::145
dns3.uminho.pt  AAAA IPv6 address = 2001:690:2280:1::65
ns02.fccn.pt    AAAA IPv6 address = 2001:690:a80:4001::200
```

```
> pt.
Server:  dns3.uminho.pt
Address:  193.137.16.65

Non-authoritative answer:
pt
        primary name server = curiosity.dns.pt
        responsible mail addr = request.dns.pt
        serial  = 2022120934
        refresh = 21600 (6 hours)
        retry   = 7200 (2 hours)
        expire  = 2592000 (30 days)
        default TTL = 300 (5 mins)

pt      nameserver = ns2.nic.fr
pt      nameserver = d.dns.pt
pt      nameserver = ns.dns.br
pt      nameserver = a.dns.pt
pt      nameserver = c.dns.pt
pt      nameserver = e.dns.pt
pt      nameserver = g.dns.pt
pt      nameserver = b.dns.pt
pt      nameserver = h.dns.pt
c.dns.pt        internet address = 204.61.216.105
h.dns.pt        internet address = 194.146.106.138
d.dns.pt        internet address = 185.39.210.1
a.dns.pt        internet address = 185.39.208.1
b.dns.pt        internet address = 194.0.25.23
e.dns.pt        internet address = 193.136.192.64
g.dns.pt        internet address = 193.136.2.226
ns.dns.br       internet address = 200.160.0.5
ns2.nic.fr      internet address = 192.93.0.4
c.dns.pt        AAAA IPv6 address = 2001:500:14:6105:ad::1
h.dns.pt        AAAA IPv6 address = 2001:67c:1010:35::53
d.dns.pt        AAAA IPv6 address = 2a04:6d82::1
a.dns.pt        AAAA IPv6 address = 2a04:6d80::1
```

```
> .
Server:  dns3.uminho.pt
Address:  193.137.16.65

Non-authoritative answer:
(root)
        primary name server = a.root-servers.net
        responsible mail addr = nstld.verisign-grs.com
        serial  = 2022120900
        refresh = 1800 (30 mins)
        retry   = 900 (15 mins)
        expire  = 604800 (7 days)
        default TTL = 86400 (1 day)

(root)  nameserver = i.root-servers.net
(root)  nameserver = e.root-servers.net
(root)  nameserver = k.root-servers.net
(root)  nameserver = d.root-servers.net
(root)  nameserver = f.root-servers.net
(root)  nameserver = h.root-servers.net
(root)  nameserver = j.root-servers.net
(root)  nameserver = c.root-servers.net
(root)  nameserver = b.root-servers.net
(root)  nameserver = a.root-servers.net
(root)  nameserver = m.root-servers.net
(root)  nameserver = l.root-servers.net
(root)  nameserver = g.root-servers.net
a.root-servers.net      internet address = 198.41.0.4
b.root-servers.net      internet address = 199.9.14.201
c.root-servers.net      internet address = 192.33.4.12
d.root-servers.net      internet address = 199.7.91.13
e.root-servers.net      internet address = 192.203.230.10
f.root-servers.net      internet address = 192.5.5.241
g.root-servers.net      internet address = 192.112.36.4
h.root-servers.net      internet address = 198.97.190.53
i.root-servers.net      internet address = 192.36.148.17
j.root-servers.net      internet address = 192.58.128.30
k.root-servers.net      internet address = 193.0.14.129
l.root-servers.net      internet address = 199.7.83.42
m.root-servers.net      internet address = 202.12.27.33
> 
```

Servidores Primários:
- uminho.pt. -> dns.uminho.pt
- pt. -> curiosity.dns.pt
- . -> a.root-servers.net

A grande diferença entre um servidor primário e um servidor secundário é que o primário carrega toda a informação da zona em causa a partir de ficheiros (base de dados) existentes em disco, enquanto que os servidores secundários obtêm toda a informação a partir do servidor primário. O DNS secundário contém apenas cópias dos ficheiros sendo que estas são read-only.

Os parâmetros temporais, e o seu significado, associados ao servidor primário, são os seguintes:

- **Serial:** Número que se dá cada vez que o DNS primário faz alguma atualização. Como este número tem de aumentar a cada atualização, geralmente este corresponde à data da atualização, de maneira que o aumento do valor deste campo seja sempre possível.

- **Refresh:** Tempo máximo que o DNS primário pode demorar a informar o DNS secundário se tem alguma atualização.

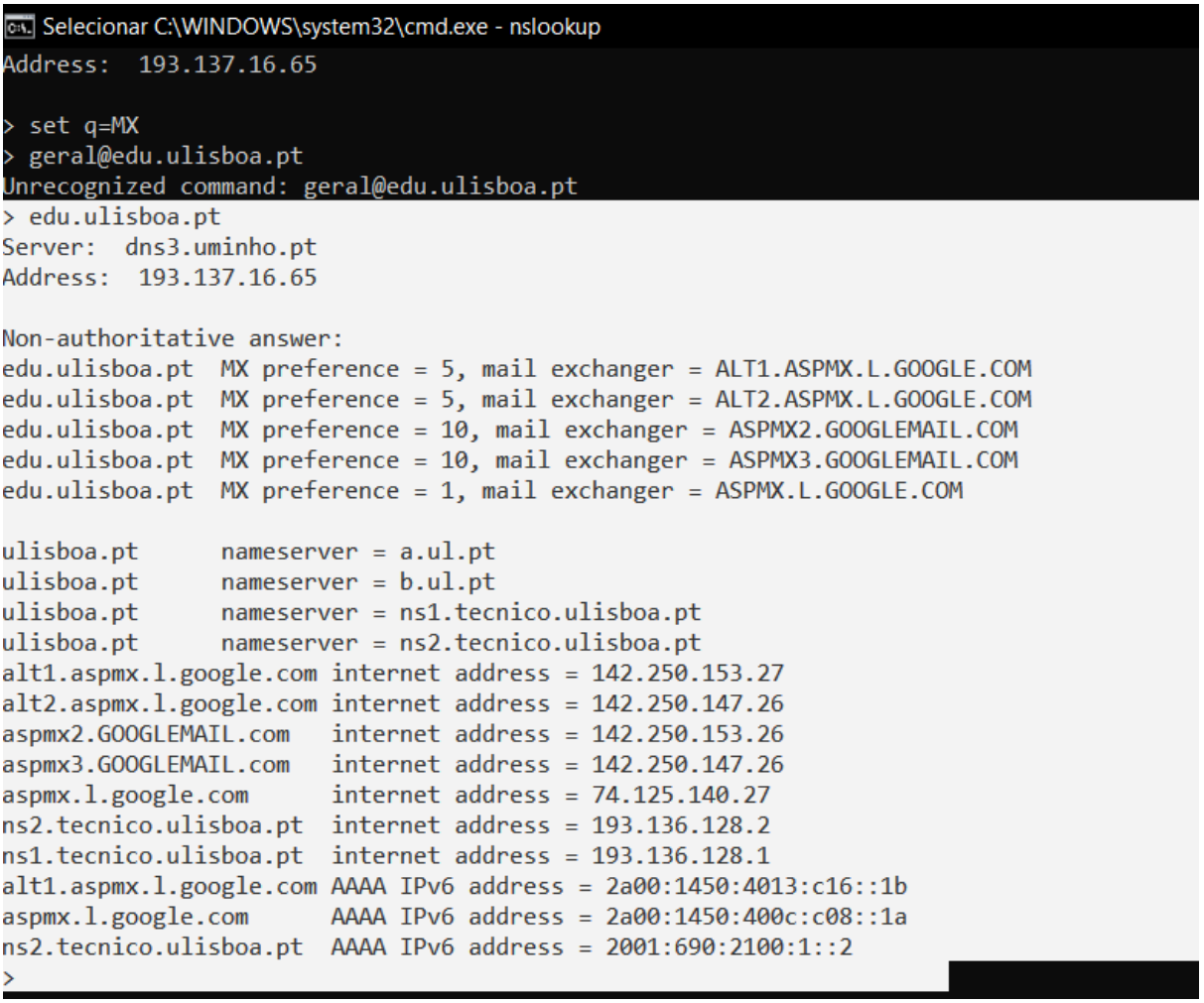
- **Retry:** Tempo que demora até tentar dar Refresh novamente, caso o DNS primário não retorne nenhuma resposta no tempo estipulado no campo Refresh.

- **Expire:** Tempo durante o qual o DNS secundário dá Retry até determinar que não consegue comunicar com o respectivo DNS primário. Quando isto acontece, o DNS secundário não retorna resposta.

- **Default TTL:** Tempo que o pacote pode permanecer na cache. Evita casos em que o pacote se perderia na rede.

Como podemos ver, os parâmetros temporais variam de servidor para servidor.

Pergunta 6) ***Usando o registo MX, diga qual(quais) o(s) servidor(s) de email do domínio edu.ulisboa.pt ? A que sistema são entregues preferencialmente as mensagens dirigidas a geral@edu.ulisboa.pt?***



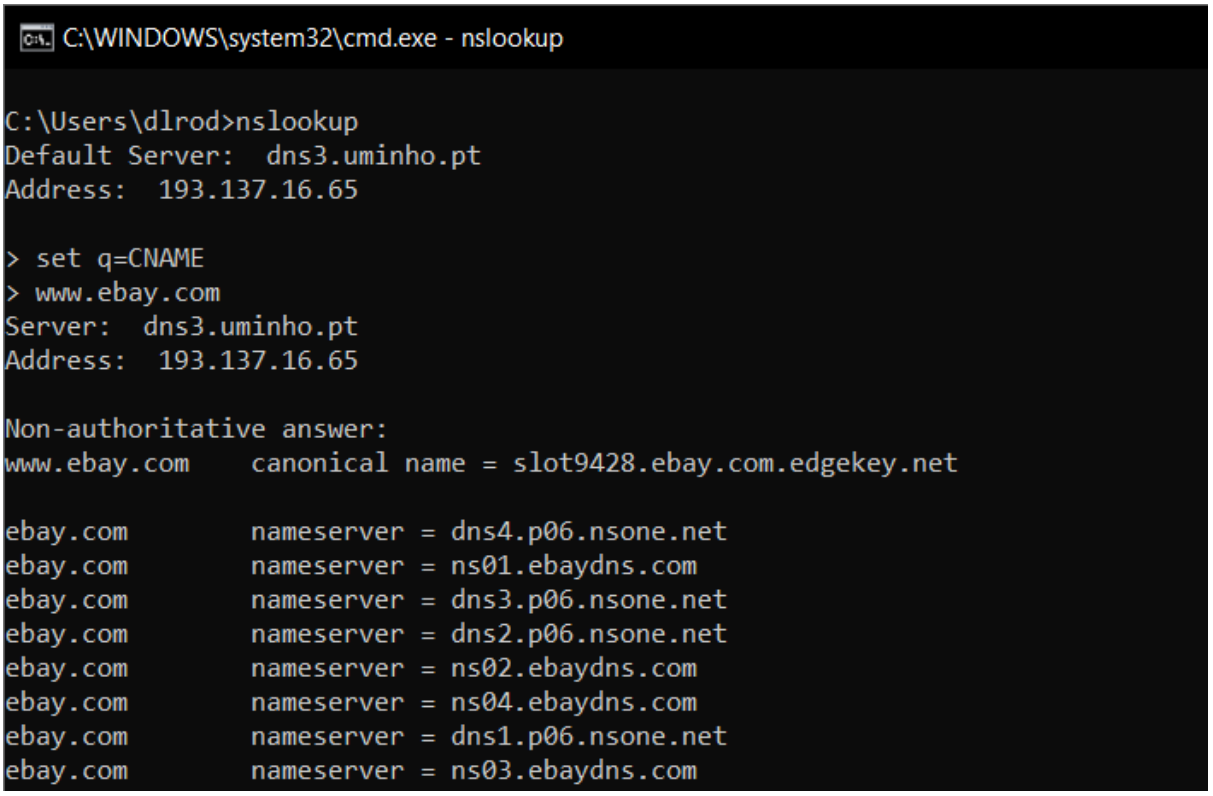
Como podemos ver os servidores de email do domínio edu.ulisboa.pt são:

edu.ulisboa.pt MX preference = 5, mail exchanger =ALT1.ASPMX.L.GOOGLE.COM
edu.ulisboa.pt MX preference = 5, mail exchanger =ALT2.ASPMX.L.GOOGLE.COM
edu.ulisboa.pt MX preference = 10, mail exchanger =ASPMX2.GOOGLEMAIL.COM
edu.ulisboa.pt MX preference = 10, mail exchanger =ASPMX3.GOOGLEMAIL.COM
edu.ulisboa.pt MX preference = 1, mail exchanger = ASPMX.L.GOOGLE.COM

É entregue ao sistema, aquele que possui o número mais pequeno de preferência, neste caso:

ASPMX.L.GOOGLE.COM

Pergunta 7) ***Usando o registo CNAME, diga qual(quais) o(s) aliases do nome www.ebay.com? O que é que isso significa?***



O registo CNAME é um registo DNS que especifica um nome alternativo para um determinado domínio.
O alias do nome www.ebay.com é:
www.ebay.com canonical name = slot9428.ebay.com.edgekey.net

Pergunta 8) ***Qual a diferença entre uma resposta adjetivada como non-authoritative answer ("não-autoritativa") e uma resposta "autoritativa" para uma determinada query?***

Uma resposta autoritativa vem de um DNS que está no domínio em que se fez a pesquisa, normalmente um DNS primário, enquanto que uma *non-authorative answer* vem de um servidor que não está nesse domínio, geralmente DNS secundários, que basicamente estão a transmitir informação em segunda mão.

4. Uso da camada de transporte por parte das aplicações

Pergunta 1) ***Preencha a seguinte tabela. Inclua todos os extratos dos outputs que lhe permitem chegar às conclusões acima.***

Comando/Aplicação	Canal Seguro	Protocolo de Transporte	Porta de atendimento
browser http://	Não	TCP	80
browser https://	Sim	TCP	443
ftp	Não	TCP	21
ping	Não	Não aplicável	Não aplicável
ssh	Sim	TCP	22
nslookup/dig	Sim	UDP	53
traceroute	Sim	Não aplicável	Não aplicável
telnet	Não	TCP	23

http:

```
Protocol: TCP (6)
Header Checksum: 0x6a1d [validation disabled]
[Header checksum status: Unverified]
Source Address: 193.137.9.178
Destination Address: 192.168.1.68
Transmission Control Protocol, Src Port: 80, Dst Port: 53627, Seq: 33581, Ack: 580, Len: 224
Source Port: 80
```

```
Line-based text data: text/html (553 lines)
\r\n
\r\n
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">\r\n
<html xmlns="http://www.w3.org/1999/xhtml">\r\n
\t<head>\r\n
\t\t<title>Servi□os de Ac□□o Social da Universidade do Minho</title>\r\n
```

Como podemos ver os dados enviados, significa que a ligação não é segura.

https:

```
  Protocol: TCP (6)
  Header Checksum: 0x0000 [validation disabled]
  [Header checksum status: Unverified]
  Source Address: 192.168.1.68
  Destination Address: 204.79.197.239
Transmission Control Protocol, Src Port: 53970, Dst Por
  Source Port: 53970
  Destination Port: 443
```

ftp:

```
Transmission Control Protocol, Src Port: 55957, Dst Port: 21, Seq: 15, Ack: 293, Len: 7
   Source Port: 55957
   Destination Port: 21
```

ping:

```
Frame 51: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface \Device\NPF_{F659D28A-8ED9-4FBC-A5C8-E5A7D428067A}, id 0
Ethernet II, Src: AlticeLa_b1:d1:6f (58:fc:20:b1:d1:6f), Dst: IntelCor_39:23:c6 (84:c5:a6:39:23:c6)
Internet Protocol Version 4, Src: 193.136.19.254, Dst: 192.168.1.68
Internet Control Message Protocol
```

Como não aparece nada após o IP, significa que não chega ao nível de transporte, logo o protocolo de transporte não é aplicável, e não existe porta de atendimento.

ssh:

```
Transmission Control Protocol, Src Port: 57877, Dst Port: 22, Seq: 1534, Ack: 1566, Len: 68
   Source Port: 57877
   Destination Port: 22
```

nslookup:

```
User Datagram Protocol, Src Port: 64157, Dst Port: 53
   Source Port: 64157
   Destination Port: 53
   Length: 42
   Checksum: 0x4853 [unverified]
   [Checksum Status: Unverified]
   [Stream index: 4]
 > [Timestamps]
   UDP payload (34 bytes)
```

Não se consegue ver os dados trocados entre o cliente e servidor, por isso é uma ligação segura.

traceroute:

```
Frame 290: 106 bytes on wire (848 bits), 106 bytes captured (848 bits) on interface \Device\NPF_{F659D28A-8ED9-4FBC-A5C8-E5A7D428067A}, id 0
Ethernet II, Src: IntelCor_39:23:c6 (84:c5:a6:39:23:c6), Dst: AlticeLa_b1:d1:6f (58:fc:20:b1:d1:6f)
Internet Protocol Version 4, Src: 192.168.1.68, Dst: 193.136.9.254
Internet Control Message Protocol
```

Como não aparece nada após o IP, significa que não chega ao nível de transporte, logo o protocolo de transporte não é aplicável, e não existe porta de atendimento.

tellnet:

```
Transmission Control Protocol, Src Port: 23, Dst Port: 61904, Seq: 1, Ack: 1, Len: 1460
   Source Port: 23
   Destination Port: 61904
   [Stream index: 0]
   [TCP Segment Len: 1460]
   Sequence Number: 1    (relative sequence number)
   Sequence Number (raw): 1432840224
   [Next Sequence Number: 1461    (relative sequence number)]
   Acknowledgment Number: 1    (relative ack number)
   Acknowledgment number (raw): 843252713
   0101 .... = Header Length: 20 bytes (5)
```

```
Telnet
   Data: \n
   Data: \r                _       __    __                                  __      \n
   Data: \r           | |     / /__  / /_______  ____ ___  ___     / /____ \n
   Data: \r           | | /| / / _ \/ / ___/ __ \/ __ `__ \/ _ \   / __/ __ \\n
   Data: \r           | |/ |/ /  __/ / /__/ /_/ / / / / / /  __/  / /_/ /_/ /\n
   Data: \r           |__/|__/\___/_/\___/\____/_/ /_/ /_/\___/   \__/\____/ \n
   Data: \r       ^^__                  _____________________                 _  _  _ \n
   Data: \r      /  - \_              / ____/  _/ ____/ ___/   _           | || || |\n
   Data: \r    <|    __<             / /_    / // /    \__ \   (_)          |_______|\n
   Data: \r    <|    \              / __/ _/ // /___ ___/ /  _              \__ ___ /\n
   Data: \r    <|     \            /_/   /___/\____//____/  (_)              |__|_|\n
   Data: \r    <|______\                                                     |_|___|\n
   Data: \r     _|____|_        ======================================       |___|_|\n
   Data: \r    (________)         freechess.org  ----  54.39.129.129        (_______)\n
   Data: \r    /________\       ======================================      /_______\ \n
   Data: \r                       (Login screen designed by Alefith)\n
   Data: \r\n
   Data: \r   ****** Welcome to the Free Internet Chess Server at freechess.org ******\n
   Data: \r\n
   Data: \rWebpage: http://www.freechess.org\n
   Data: \rHead admin : mattuc   Complaints to : complaints@freechess.org\n
   Data: \rServer location: freechess.org   Server version : 1.25.20\n
   Data: \r\n
   Data: \r      If you are not a registered player, enter guest or a unique ID.\n
   Data: \r            (If your return key does not work, use
```

Como podemos ver os dados enviados, significa que a ligação não é segura.

Pergunta 2) ***Comente as principais diferenças entre os protocolos TCP e UDP. Relacione-as com as experiências realizadas onde observou os campos dos cabeçalhos respetivos e o overhead protocolar. Em particular, identifique os campos do TCP responsáveis pelo controlo de fluxo, ordenação e fiabilidade do protocolo.***

Tanto o protocolo TCP como UDP são protocolos fim-a-fim.
No entanto, o protocolo UDP, não é fiável, pois, quando ocorre um erro, este descarta o pacote, não há retransmissão.
No caso do TCP, apesar de fornecer uma transmissão de dados mais lenta, é um protocolo fiável, pois faz controlo de fluxo, de forma a não sobrecarregar o receptor, controlo de congestão, de forma a não sobrecarregar a rede, e controlo de erros, por time-out, se não chegar confirmação da chegada, retransmite.
Ao contrário do UDP, o TCP não pode ser usado para serviços broadcasting e multicasting.

Conclusão:

Com este relatório aprendemos mais sobre os protocolos de transporte, TCP e UDP, as suas diferenças e aplicações, e quais as vantagens e desvantagens de cada um.

Aprendemos também que a internet nem sempre está segura de ataques, e que é necessário tomar atenção se estamos numa ligação segura ou não.

Também ficamos a saber mais sobre a parte aplicacional das redes.