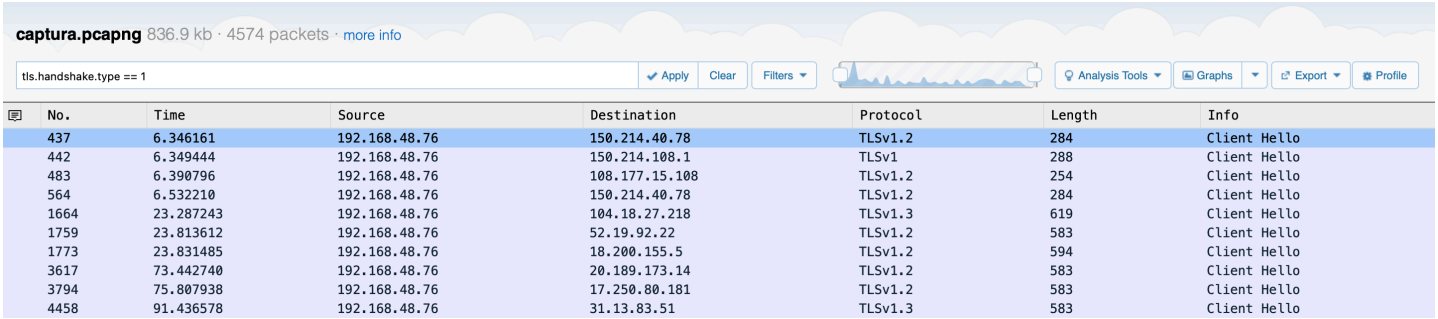
Práctica Evaluable 2

Jacobo Elicha Garrucho

Ejercicio 1: Análisis de protocolos

1.1 ¿Cuántas conexiones TLS se establecen y con cuantos servidores diferentes? ¿Qué filtro de Wireshark te permite responder fácilmente a esta pregunta?

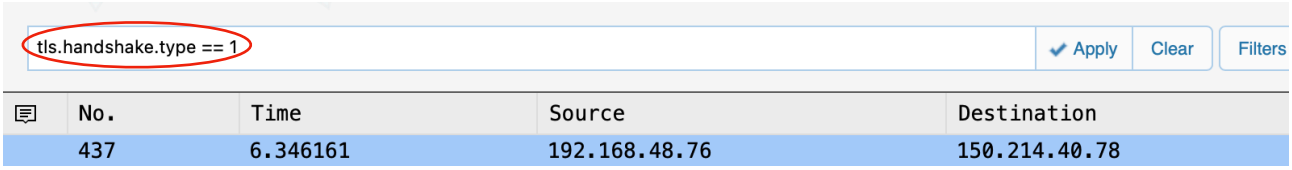
Ya sabemos que cada vez que se establece una conexión TLS entre un cliente y un servidor, lo primero que sucede es que el cliente le manda la trama ClientHello (trama de TLS con el tipo 1) al servidor. Como podemos observar hay 10 de estas tramas, con lo que se establecen **10 conexiones TLS** con **9 servidores distintos**.



No.	Time	Source	Destination	Protocol	Length	Info
437	6.346161	192.168.48.76	150.214.40.78	TLSv1.2	284	Client Hello
442	6.349444	192.168.48.76	150.214.108.1	TLSv1	288	Client Hello
483	6.390796	192.168.48.76	108.177.15.108	TLSv1.2	254	Client Hello
564	6.532210	192.168.48.76	150.214.40.78	TLSv1.2	284	Client Hello
1664	23.287243	192.168.48.76	104.18.27.218	TLSv1.3	619	Client Hello
1759	23.813612	192.168.48.76	52.19.92.22	TLSv1.2	583	Client Hello
1773	23.831485	192.168.48.76	18.200.155.5	TLSv1.2	594	Client Hello
3617	73.442740	192.168.48.76	20.189.173.14	TLSv1.2	583	Client Hello
3794	75.807938	192.168.48.76	17.250.80.181	TLSv1.2	583	Client Hello
4458	91.436578	192.168.48.76	31.13.83.51	TLSv1.3	583	Client Hello

(También podríamos haber estudiado las tramas ServerHello)

Como se puede observar en la captura, el filtro que usamos es **tls.handshake.type == 1**, puesto que estamos buscando las tramas del protocolo TLS, durante el handshake, y en específico las del ClientHello que sabemos tienen tipo 1.



No.	Time	Source	Destination
437	6.346161	192.168.48.76	150.214.40.78

1.2 ¿Qué versión de TLS se utiliza en la conexión con el host 150.214.108.1? ¿Y con el host 150.214.40.78?

Para ver que versión de TLS se utiliza con el host 150.214.108.1 podemos simplemente buscarlo en la captura de arriba y ver en la columna de *Protocol* cual aparece, o también podemos filtrar (en el caso de que hubiesen muchos) por dirección IP usando el filtro ip.addr == 150.214.108.1. En ambos casos vemos que para esta IP la versión es la **v1**, como se puede apreciar en la captura.

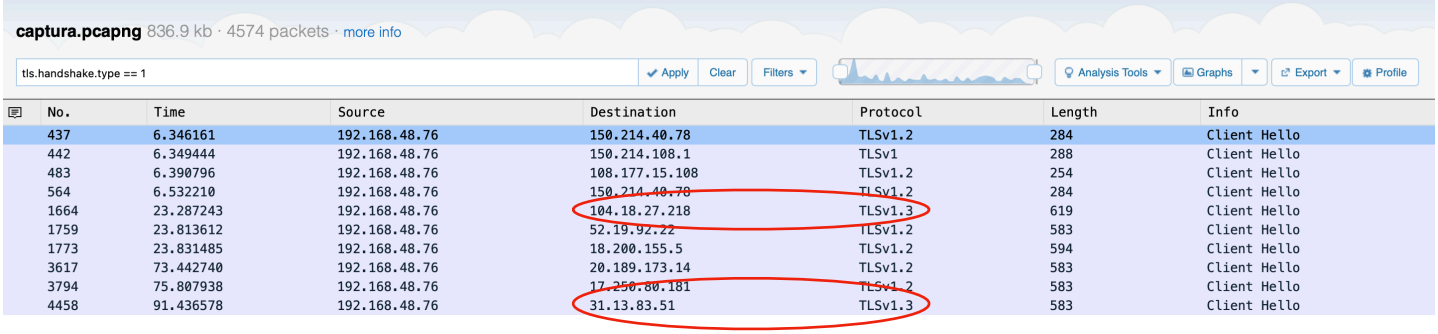
Destination	Protocol	Length	Info
150.214.108.1	TLSv1	288	Client Hello

Análogamente vemos que para el host 150.214.40.78 se usa la versión **v1.2.**

Destination	Protocol	Length	Info
150.214.40.78	TLSv1.2	284	Client Hello

1.3 ¿Existe alguna conexión TLS 1.3? En caso afirmativo indica con qué host.

Como podemos observar en la primera captura, existen dos conexiones con TLSv1.3. Una con el host 104.18.27.218 y otra con el host 31.13.83.51.



No.	Time	Source	Destination	Protocol	Length	Info
437	6.346161	192.168.48.76	150.214.40.78	TLSv1.2	284	Client Hello
442	6.349444	192.168.48.76	150.214.108.1	TLSv1	288	Client Hello
483	6.390796	192.168.48.76	108.177.15.108	TLSv1.2	254	Client Hello
564	6.532210	192.168.48.76	150.214.40.78	TLSv1.2	284	Client Hello
1664	23.287243	192.168.48.76	104.18.27.218	TLSv1.3	619	Client Hello
1759	23.813612	192.168.48.76	52.19.92.22	TLSv1.2	583	Client Hello
1773	23.831485	192.168.48.76	18.200.155.5	TLSv1.2	594	Client Hello
3617	73.442740	192.168.48.76	20.189.173.14	TLSv1.2	583	Client Hello
3794	75.807938	192.168.48.76	17.250.80.181	TLSv1.2	583	Client Hello
4458	91.436578	192.168.48.76	31.13.83.51	TLSv1.3	583	Client Hello

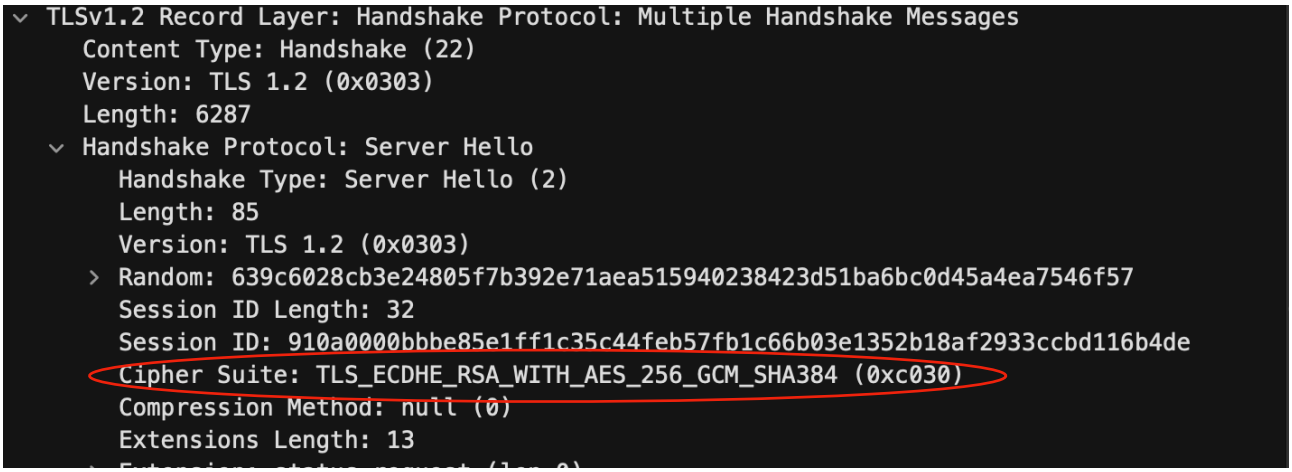
1.4 En la conexión con el servidor 20.189.173.14, ¿qué suites de cifrado se ofertan al servidor? ¿cuál es la elegida para establecer la conexión?

Como podemos observar, se ofertan:

```
Cipher Suites Length: 34
Cipher Suites (17 suites)
    Cipher Suite: TLS_AES_128_GCM_SHA256 (0x1301)
    Cipher Suite: TLS_AES_256_GCM_SHA384 (0x1302)
    Cipher Suite: TLS_CHACHA20_POLY1305_SHA256 (0x1303)
    Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 (0xc02c)
    Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 (0xc02b)
    Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 (0xc024)
    Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 (0xc023)
    Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA (0xc00a)
    Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA (0xc009)
    Cipher Suite: TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256 (0xcca9)
    Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (0xc030)
    Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f)
    Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 (0xc028)
    Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 (0xc027)
    Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xc014)
    Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (0xc013)
    Cipher Suite: TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 (0xcca8)
```

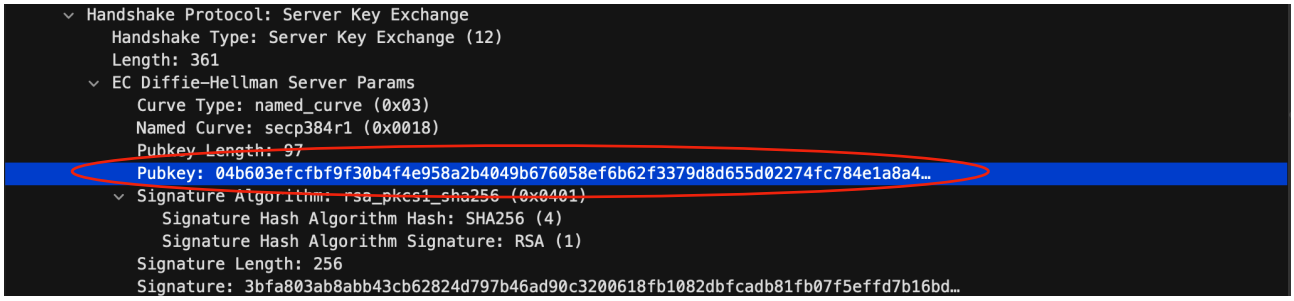
(Esto lo podemos ver entrando en la trama de ClientHello con destino al servidor indicado)

Para ver cual es la elegida debemos mirar la correspondiente trama de respuesta ServerHello. [Dejo de usar Cloudshark y comienzo a trabajar con Wireshark, por eso el cambio de tonalidades.]

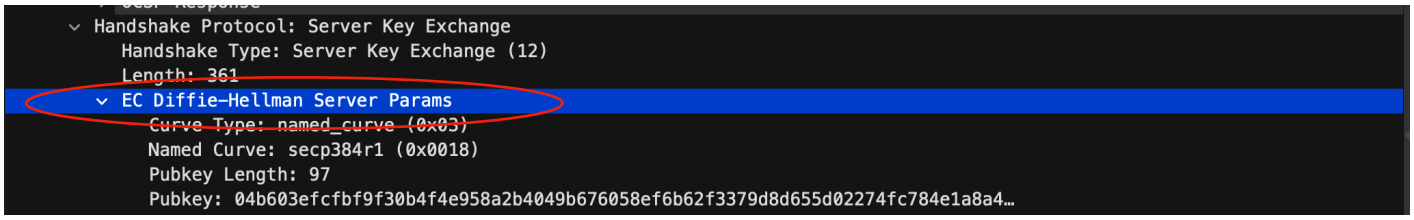


1.5 ¿Cuál es la clave pública del servidor 20.189.173.14? ¿De qué tipo de clave se trata (i.e., para qué algoritmo se utiliza)?

La clave pública es:



El algoritmo es:



Ejercicio 2: Cortafuegos

2.1 ¿Qué servicios pueden ser accedidos desde internet?

Se puede acceder al servidor con ip 192.168.3.2 mediante http y https.

[1]:

```
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80 -j DNAT --to 192.168.3.2:80
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 443 -j DNAT --to 192.168.3.2:443
```

2.2 ¿Existe una zona desmilitarizada (DMZ) en la red?

Si, desde internet solo se puede acceder a la maquina con ip 192.168.3.2 (ya sea mediante http o https) [1], que será el servidor que se encuentre en la zona desmilitarizada. Posteriormente vemos que solo se aceptan los paquetes en la red interna de las maquinas con ip comprendida en 192.168.10.0/24 [2]. Luego en nuestra DMZ se encuentra la maquina 192.168.3.2.

[2]:

```
iptables -A INPUT -s 192.168.10.0/24 -i eth1 -j ACCEPT
```

2.3 ¿Los hosts de la red 192.168.10.0/24 pueden acceder a Internet?

Si tienen acceso a internet puesto que se les proporciona una ip pública (de forma dinámica) con [3]. Pero recordemos que la política por defecto es rechazar todo tráfico entrante o saliente, y en ningún momento se establece una regla que permita enviar o recibir paquetes para los hosts de la red 192.168.10.0/24 a través de eth0 (internet).

[3]:

```
iptables -t nat -A POSTROUTING -s 192.168.10.0/24 -o eth0 -j MASQUERADE
```

2.4 ¿Pueden los hosts de la red 192.168.10.0/24 recibir conexiones desde internet?

No, puesto que las Políticas por Defecto establecen todo a DROP (prohibitiva luego no permiten por defecto entrada ni salida de paquetes) y no hay ninguna regla que permita la entrada de paquetes desde eth0 a la red 192.168.10.0/24.

2.5 ¿Pueden los hosts de la red 192.168.10.0/24 recibir conexiones de la red 192.168.3.0/24?

No, todo el trafico que pueden recibir las maquinas de la red 192.168.10.0/24 son las que provienen de la misma red (por la interfaz eth1) por la siguiente regla [4].

[4]:

```
iptables -A INPUT -s 192.168.10.0/24 -i eth1 -j ACCEPT
```

----- IPTABLES.SH -----

```
#!/bin/sh
# ----------------------
# IPTABLES script
# ----------------------

#1) Esto es un FLUSH de reglas, 'limpia' las reglas que se puedan
encontrar ya escritas para partir de cero, esto es elemental.
iptables -F
iptables -X
iptables -Z
iptables -t nat -F

#2)Establece nuestras Politicas por Defecto, en este caso es
restrictivo puesto que por defecto todo trafico se rechaza
(incluyendo al Router).
#Esto quiere decir que, cualquier trafico que no entre por alguna
de las reglas que se escriban posteriormente será rechazado.
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP
iptables -t nat -P PREROUTING DROP
iptables -t nat -P POSTROUTING DROP

#3)Se agregan estas reglas en la tabla nat que son utilizadas para
modificar, de los paquetes que entran por eth0, la ip de destino a
192.168.3.2 con
#sus respectivos puertos, en los casos donde el puerto sea el 80
(protocolo HTTP) o bien el 443 (protocolo HTTPS), realizando esto
antes de ser enrutado el paquete
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80 -j DNAT -
-to 192.168.3.2:80
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 443 -j DNAT -
-to 192.168.3.2:443

#4)Se agrega una regla que permite (ACCEPT) la entrada de paquetes
que vengan de las ips 192.168.10.0/24 a traves de la interfaz de
red eth1
iptables -A INPUT -s 192.168.10.0/24 -i eth1 -j ACCEPT
```

```
#5)Estas reglas hacen NAT (proprocionan ip pública de forma dinamica) si la ip origen es una de las ips 192.168.10.0/24 o 192.168.3.0/24 y sale por eth0
iptables -t nat -A POSTROUTING -s 192.168.10.0/24 -o eth0 -j MASQUERADE
iptables -t nat -A POSTROUTING -s 192.168.3.0/24 -o eth0 -j MASQUERADE

#6)Esta sentencia activa el IP Forwarding, para que el equipo permita que pasen paquetes de una interfaz de red a otra
echo 1 > /proc/sys/net/ipv4/ip_forward


#7)Se agregan dos reglas a la tabla filter, la primera establece el reenvio (forwarding) de paquetes que entren por la interfaz de red eth1 con estado NEW,
#ESTABLISHED O RELATED a traves de eth2, y la siguiente establece el reenvio de lo que entra por eth2 con estado ESTABLISHED O RELATED a traves de eth1
iptables -t filter -A FORWARD -i eth1 -o eth2 -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT
iptables -t filter -A FORWARD -i eth2 -o eth1 -m state --state ESTABLISHED,RELATED -j ACCEPT
```

Ejercicio 3: Certificados y Correo electrónico