



# SERVIDORES WEB DE ALTAS PRESTACIONES

---

## TEMA 5

**Autor**

Miguel Ángel Pérez Díaz



Escuela Técnica Superior de Ingenierías Informática y  
de Telecomunicación

—  
Granada, 2020

## T5. WIRESHARK.

***Instalar Wireshark y observar cómo fluye el tráfico de red en el balanceador de la máquina M3 mientras se le hacen peticiones HTTP y HTTPS. Ejecuta al menos 3 peticiones al balanceador.***

***Realiza un análisis de una sesión TCP (establecer conexión y cierre) de peticiones HTTP y HTTPS y escribe tus propias conclusiones. Puedes ilustrarlo con capturas de pantalla.***

Para esta tarea ha sido necesario instalar la herramienta Wireshark para monitorizar el tráfico de la red a través de adaptador Solo-Anfitrión por el que están conectados el balanceador y los servidores. A continuación, se realizarán 3 peticiones HTTP y HTTPS a través de mi máquina anfitriona al balanceador y analizaremos el tráfico resultante mediante Wireshark.

Inicialmente se han enviado 3 peticiones HTTP a la dirección IP del balanceador: *192.168.56.104* desde mi máquina anfitriona *192.168.56.1*, obteniéndose el siguiente resultado:

*VirtualBox Host-Only Network							
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help							
Apply a display filter ... <Ctrl-/>							
No.	Time	Source	Destination	Protocol	Length	Info	
103	52.525360	192.168.56.1	192.168.56.104	TCP	66	63313 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1	
104	52.525779	192.168.56.104	192.168.56.1	TCP	66	80 → 63313 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM=1 WS=64	
105	52.525885	192.168.56.1	192.168.56.104	TCP	54	63313 → 80 [ACK] Seq=1 Ack=1 Win=2102272 Len=0	
106	52.526182	192.168.56.1	192.168.56.104	HTTP	132	GET / HTTP/1.1	
107	52.526465	192.168.56.104	192.168.56.1	TCP	60	80 → 63313 [ACK] Seq=1 Ack=79 Win=64192 Len=0	
108	52.528763	192.168.56.104	192.168.56.1	HTTP	397	HTTP/1.1 200 OK (text/html)	
109	52.536298	192.168.56.1	192.168.56.104	TCP	54	63313 → 80 [FIN, ACK] Seq=79 Ack=344 Win=2102016 Len=0	
110	52.536733	192.168.56.104	192.168.56.1	TCP	60	80 → 63313 [FIN, ACK] Seq=344 Ack=80 Win=64192 Len=0	
111	52.536814	192.168.56.1	192.168.56.104	TCP	54	63313 → 80 [ACK] Seq=80 Ack=345 Win=2102016 Len=0	
112	52.974683	192.168.56.12	192.168.0.1	TCP	74	[TCP Retransmission] 33410 → 53 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=3439688621 TSecr=0 WS=64	
113	53.126887	192.168.56.1	192.168.56.104	TCP	66	63314 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1	
114	53.127458	192.168.56.104	192.168.56.1	TCP	66	80 → 63314 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM=1 WS=64	
115	53.127593	192.168.56.1	192.168.56.104	TCP	54	63314 → 80 [ACK] Seq=1 Ack=1 Win=2102272 Len=0	
116	53.127929	192.168.56.1	192.168.56.104	HTTP	132	GET / HTTP/1.1	
117	53.128171	192.168.56.104	192.168.56.1	TCP	60	80 → 63314 [ACK] Seq=1 Ack=79 Win=64192 Len=0	
118	53.131390	192.168.56.104	192.168.56.1	HTTP	397	HTTP/1.1 200 OK (text/html)	
119	53.140736	192.168.56.1	192.168.56.104	TCP	54	63314 → 80 [FIN, ACK] Seq=79 Ack=344 Win=2102016 Len=0	
120	53.141380	192.168.56.104	192.168.56.1	TCP	60	80 → 63314 [FIN, ACK] Seq=344 Ack=80 Win=64192 Len=0	
121	53.141469	192.168.56.1	192.168.56.104	TCP	54	63314 → 80 [ACK] Seq=80 Ack=345 Win=2102016 Len=0	
122	53.229563	PcsCompu_c9:1f:07	Broadcast	ARP	60	Who has 192.168.56.101? Tell 192.168.56.12	
123	53.768727	192.168.56.1	192.168.56.104	TCP	66	63315 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1	
124	53.769218	192.168.56.104	192.168.56.1	TCP	66	80 → 63315 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM=1 WS=64	
125	53.769355	192.168.56.1	192.168.56.104	TCP	54	63315 → 80 [ACK] Seq=1 Ack=1 Win=2102272 Len=0	
126	53.769706	192.168.56.1	192.168.56.104	HTTP	132	GET / HTTP/1.1	
127	53.770010	192.168.56.104	192.168.56.1	TCP	60	80 → 63315 [ACK] Seq=1 Ack=79 Win=64192 Len=0	
128	53.773035	192.168.56.104	192.168.56.1	HTTP	397	HTTP/1.1 200 OK (text/html)	
129	53.781662	192.168.56.1	192.168.56.104	TCP	54	63315 → 80 [FIN, ACK] Seq=79 Ack=344 Win=2102016 Len=0	
130	53.782339	192.168.56.104	192.168.56.1	TCP	60	80 → 63315 [FIN, ACK] Seq=344 Ack=80 Win=64192 Len=0	
131	53.782426	192.168.56.1	192.168.56.104	TCP	54	63315 → 80 [ACK] Seq=80 Ack=345 Win=2102016 Len=0	

Se puede observar como en cada petición realizada el balanceador se realiza el triple handshake, es decir, en el primer paso mi máquina anfitriona que desea establecer la conexión envía al balanceador un paquete SYN. Después cuando el balanceador ha recibido el segmento, confirma el establecimiento de la conexión mediante el envío de un paquete SYN-ACK, para terminar finalmente mi máquina anfitriona confirma la recepción del segmento SYN-ACK mediante el envío de un paquete ACK propio.

Todo el proceso mediante el puerto de HTTP : 80.

Una vez establecida la conexión TCP se procede a las peticiones al balanceador mediante una petición GET, el balanceador confirma que ha recibo la petición con un ACK para posteriormente realizar el envío de datos, en este caso texto HTML.

Finalmente, la máquina anfitriona confirma que lo recibió con un ACK e indica el fin de la sesión, el balanceador envía de vuelta la misma petición indicando que ha recibo la solicitud y que también cerrará la sesión. Por último, el cliente devuelve un ACK como recibo de confirmación y finaliza la conexión para dicha solicitud.

Posteriormente se han enviado 3 peticiones HTTPS a la dirección IP del balanceador: *192.168.56.104* desde mi máquina anfitriona *192.168.56.1*, obteniéndose el siguiente resultado:

VirtualBox Host-Only Network						
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help						
Apply a display filter ... <Ctrl+>						
No.	Time	Source	Destination	Protocol	Length	Info
133	54.709683	192.168.56.1	192.168.56.104	TCP	66	63316 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
134	54.710139	192.168.56.104	192.168.56.1	TCP	66	443 → 63316 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM=1 WS=64
135	54.710269	192.168.56.1	192.168.56.104	TCP	54	63316 → 443 [ACK] Seq=1 Ack=1 Win=2102272 Len=0
136	54.737690	192.168.56.1	192.168.56.104	TLShv1.2	571	Client Hello
137	54.738100	192.168.56.104	192.168.56.1	TCP	60	443 → 63316 [ACK] Seq=1 Ack=518 Win=64128 Len=0
138	54.740539	192.168.56.104	192.168.56.1	TLShv1.2	1509	Server Hello, Certificate, Server Key Exchange, Server Hello Done
139	54.742517	192.168.56.1	192.168.56.104	TLShv1.2	147	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
140	54.743436	192.168.56.104	192.168.56.1	TLShv1.2	105	Change Cipher Spec, Encrypted Handshake Message
141	54.743590	192.168.56.1	192.168.56.104	TCP	54	63316 → 443 [ACK] Seq=611 Ack=1507 Win=2102272 Len=0
142	54.744051	192.168.56.1	192.168.56.104	TLShv1.2	161	Application Data
143	54.746985	192.168.56.104	192.168.56.1	TLShv1.2	426	Application Data
144	54.754909	192.168.56.1	192.168.56.104	TLShv1.2	85	Encrypted Alert
145	54.755485	192.168.56.104	192.168.56.1	TCP	60	443 → 63316 [FIN, ACK] Seq=1879 Ack=749 Win=64128 Len=0
146	54.755565	192.168.56.1	192.168.56.104	TCP	54	63316 → 443 [ACK] Seq=749 Ack=1880 Win=2102016 Len=0
147	54.756000	192.168.56.1	192.168.56.104	TCP	54	63316 → 443 [FIN, ACK] Seq=749 Ack=1880 Win=2102016 Len=0
148	54.756340	192.168.56.104	192.168.56.1	TCP	60	443 → 63316 [ACK] Seq=1880 Ack=750 Win=64128 Len=0
149	55.277744	PcsCompu_c9:1f:07	Broadcast	ARP	60	Who has 192.168.56.101? Tell 192.168.56.12
150	55.385767	192.168.56.1	192.168.56.104	TCP	66	63317 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
151	55.386209	192.168.56.104	192.168.56.1	TCP	66	443 → 63317 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM=1 WS=64
152	55.386390	192.168.56.1	192.168.56.104	TCP	54	63317 → 443 [ACK] Seq=1 Ack=1 Win=2102272 Len=0
153	55.409872	192.168.56.1	192.168.56.104	TLShv1.2	571	Client Hello
154	55.410269	192.168.56.104	192.168.56.1	TCP	60	443 → 63317 [ACK] Seq=1 Ack=518 Win=64128 Len=0
155	55.412349	192.168.56.104	192.168.56.1	TLShv1.2	1509	Server Hello, Certificate, Server Key Exchange, Server Hello Done
156	55.414079	192.168.56.1	192.168.56.104	TLShv1.2	147	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
157	55.414895	192.168.56.104	192.168.56.1	TLShv1.2	105	Change Cipher Spec, Encrypted Handshake Message
158	55.415029	192.168.56.1	192.168.56.104	TCP	54	63317 → 443 [ACK] Seq=611 Ack=1507 Win=2102272 Len=0
159	55.415477	192.168.56.1	192.168.56.104	TLShv1.2	161	Application Data
160	55.418261	192.168.56.104	192.168.56.1	TLShv1.2	426	Application Data
161	55.426644	192.168.56.1	192.168.56.104	TLShv1.2	85	Encrypted Alert
162	55.427179	192.168.56.104	192.168.56.1	TCP	60	443 → 63317 [FIN, ACK] Seq=1879 Ack=749 Win=64128 Len=0

163	55.427250	192.168.56.1	192.168.56.104	TCP	54 63317 → 443 [ACK] Seq=749 Ack=1880 Win=2102016 Len=0
164	55.427665	192.168.56.1	192.168.56.104	TCP	54 63317 → 443 [FIN, ACK] Seq=749 Ack=1880 Win=2102016 Len=0
165	55.427979	192.168.56.104	192.168.56.1	TCP	60 443 → 63317 [ACK] Seq=1880 Ack=750 Win=64128 Len=0
166	55.826854	192.168.56.12	192.168.0.1	TCP	74 33414 → 53 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=3439691473 TSecr=0 WS=64
167	56.143757	192.168.56.1	192.168.56.104	TCP	66 63318 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
168	56.144144	192.168.56.104	192.168.56.1	TCP	66 443 → 63318 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM=1 WS=64
169	56.144253	192.168.56.1	192.168.56.104	TCP	54 63318 → 443 [ACK] Seq=1 Ack=1 Win=2102272 Len=0
170	56.170275	192.168.56.1	192.168.56.104	TLSv1.2	571 Client Hello
171	56.170644	192.168.56.104	192.168.56.1	TCP	60 443 → 63318 [ACK] Seq=1 Ack=518 Win=64128 Len=0
172	56.172781	192.168.56.104	192.168.56.1	TLSv1.2	1509 Server Hello, Certificate, Server Key Exchange, Server Hello Done
173	56.174536	192.168.56.1	192.168.56.104	TLSv1.2	147 Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
174	56.175342	192.168.56.104	192.168.56.1	TLSv1.2	105 Change Cipher Spec, Encrypted Handshake Message
175	56.175464	192.168.56.1	192.168.56.104	TCP	54 63318 → 443 [ACK] Seq=611 Ack=1507 Win=2102272 Len=0
176	56.175875	192.168.56.1	192.168.56.104	TLSv1.2	161 Application Data
177	56.178503	192.168.56.104	192.168.56.1	TLSv1.2	426 Application Data
178	56.186062	192.168.56.1	192.168.56.104	TLSv1.2	85 Encrypted Alert
179	56.186736	192.168.56.104	192.168.56.1	TCP	60 443 → 63318 [FIN, ACK] Seq=1879 Ack=749 Win=64128 Len=0
180	56.186809	192.168.56.1	192.168.56.104	TCP	54 63318 → 443 [ACK] Seq=749 Ack=1880 Win=2102016 Len=0
181	56.187217	192.168.56.1	192.168.56.104	TCP	54 63318 → 443 [FIN, ACK] Seq=749 Ack=1880 Win=2102016 Len=0
182	56.187567	192.168.56.104	192.168.56.1	TCP	60 443 → 63318 [ACK] Seq=1880 Ack=750 Win=64128 Len=0

Al seguir siendo una conexión TCP se realiza de nuevo el triple handshake explicado anteriormente pero en este caso al puerto HTTPS : 443.

Después se realiza otro handshake relativo al protocolo de seguridad TSL (TLS Handshake), implica la configuración del identificador de sesión, la versión del protocolo TLS, la negociación del conjunto de cifrado, la autenticación del certificado de los pares y el intercambio de claves criptográficas entre pares.

El primer mensaje en TLS Handshake: “Client Hello” es el mensaje de saludo del cliente que el cliente envía para iniciar una sesión con el balanceador.

Después el balanceador envía de vuelta un mensaje de saludo al cliente con la siguiente estructura:

- *Server Hello*
- *Certificado del balanceador que contiene su clave pública*
- *Server Key Exchange: se envía cuando la clave pública presente en el certificado del servidor no es adecuada para el intercambio de claves o si el conjunto de cifrado establece una restricción que requiere una clave temporal.*
- *Server Hello Done: servidor está listo y está esperando la respuesta del cliente.*

El cliente envía la siguiente respuesta al balanceador:

- *Cliente Key Exchange : intercambio de clave del cliente*
- *Cambiar especificaciones de cifrado (Change Cipher Sec): notifica al balanceador que todos los mensajes futuros se cifrarán utilizando el algoritmo y las claves que se acaban de negociar.*
- *Encrypted Handshake Message: indica que la negociación TLS se ha completado para el cliente.*

Posteriormente el balanceador devuelve al cliente una respuesta con *Change Cipher Sec* y *Encrypted Handshake Message*.

El cliente envía una confirmación ACK y se procede al envío de información: Application Data (Una vez que todo el Handshake TLS se completa con éxito y se validan los pares, las aplicaciones en los pares pueden comenzar a comunicarse entre sí)

Finalmente volvemos al protocolo TCP y terminamos la conexión al igual que se hizo con HTTP.