PROTOCOLO HTTP – LABORATORIO WIRESHARK

Grupo: Julio Prado, Martín Gómez, Daniel Plazas

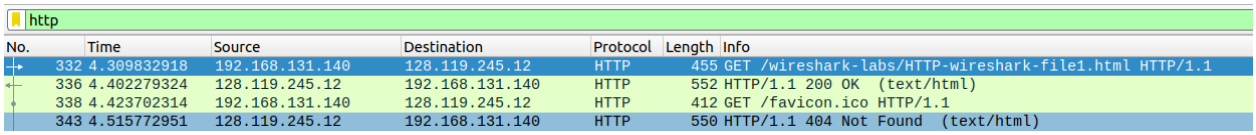
En el presente laboratorio trabajaremos varios aspectos del protocolo HTTP, a saber:

- Solicitudes básicas con GET
- GET condicional
- Documentos HTML con objetos embebidos
- Autenticación en HTTP

a. Solicitudes básicas con GET

En esta primera parte del laboratorio se accederá a un documento HTML muy sencillo, sin objetos embebidos. Para ello:

- Inicie su navegador web.
- Inicie Wireshark y comience la captura de paquetes. En el campo de filtro escriba `http` para filtrar solamente el tráfico HTTP.
- Navegue a la siguiente dirección: http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file1.html
- El navegador mostrará una página web muy sencilla, de una sola línea.
- Detenga la captura en Wireshark.



Busque en la captura el paquete que contiene la orden GET. Seleccione, dé clic derecho y escoja la opción `Follow -> TCP Stream`. Aparecerá una ventana donde se verá todo el texto intercambiado por el cliente y el servidor durante la conversación. Note que el texto enviado por el cliente aparecerá en rojo, y las respuestas del servidor en azul. Conteste las siguientes preguntas:

```
GET /wireshark-labs/HTTP-wireshark-file1.html HTTP/1.1
Host: gaia.cs.umass.edu
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:125.0) Gecko/20100101 Firefox/125.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Upgrade-Insecure-Requests: 1

HTTP/1.1 200 OK
Date: Tue, 07 May 2024 13:39:40 GMT
Server: Apache/2.4.6 (CentOS) OpenSSL/1.0.2k-fips PHP/7.4.33 mod_perl/2.0.11 Perl/v5.16.3
Last-Modified: Tue, 07 May 2024 05:59:01 GMT
ETag: "80-617d6e1f6bbe0"
Accept-Ranges: bytes
Content-Length: 128
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=UTF-8

<html>
Congratulations.  You've downloaded the file
http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file1.html!
</html>
```

1. ¿Cuál versión de HTTP emplea su navegador? ¿Cuál versión emplea el servidor?

   Navegador: 1.1
   Servidor: 1.1

2. ¿Cuáles idiomas (si aparecen) puede aceptar su navegador del servidor?

   en-US,en

3. ¿Cuál es la dirección IP de su computador? ¿Y del servidor `gaia.cs.umass.edu`?

   computador personal: 192.168.131.140
   gaia:128.119.245.12

4. ¿Cuál es el código de estado que regresa el servidor a su navegador?

   200 OK

5. ¿Cuándo fue modificado por última vez el archivo al cual accedió en el servidor?

   Last-Modified: Tue, 07 May 2024 05:59:01 GMT

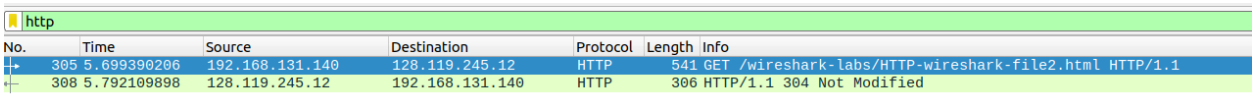
6. ¿Cuántos bytes de contenido fueron retornados por el servidor?

   Content-Length: 128

b. GET condicional

En la clase, se vio que la mayoría de los navegadores prefieren hacer un GET condicional cuando se está accediendo a una página web, para poder aprovechar el caché del navegador. En esta parte de la práctica se podrá apreciar este mecanismo en acción. Para ello:

- Borre el caché de su navegador.
- Inicie la captura en Wireshark. Emplee el filtro `http`.
- Navegue a la siguiente página: http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file2.html . Se desplegará una página web muy sencilla de 5 líneas.
- Presione el botón de refresco en el navegador, para volver a cargar la página.
- Detenga la captura en Wireshark. Se podrán apreciar dos solicitudes GET en la captura.




```
GET /wireshark-labs/HTTP-wireshark-file2.html HTTP/1.1
Host: gaia.cs.umass.edu
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:125.0) Gecko/20100101 Firefox/125.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Upgrade-Insecure-Requests: 1

HTTP/1.1 200 OK
Date: Tue, 07 May 2024 13:49:18 GMT
Server: Apache/2.4.6 (CentOS) OpenSSL/1.0.2k-fips PHP/7.4.33 mod_perl/2.0.11 Perl/v5.16.3
Last-Modified: Tue, 07 May 2024 05:59:01 GMT
ETag: "173-617d6e1f6b410"
Accept-Ranges: bytes
Content-Length: 371
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=UTF-8


<html>

Congratulations again!  Now you've downloaded the file lab2-2.html. <br>
This file's last modification date will not change.  <p>
Thus  if you download this multiple times on your browser, a complete copy <br>
will only be sent once by the server due to the inclusion of the IN-MODIFIED-SINCE<br>
field in your browser's HTTP GET request to the server.

</html>
```

```
▸ Frame 315: 455 bytes on wire (3640 bits), 455 bytes captured (3640 bits) on interface enp1s0f0, id 0
▸ Ethernet II, Src: IntelCor_c3:6e:04 (b4:96:91:c3:6e:04), Dst: Cisco_61:8d:5f (5c:5a:c7:61:8d:5f)
▸ Internet Protocol Version 4, Src: 192.168.131.140, Dst: 128.119.245.12
▸ Transmission Control Protocol, Src Port: 55668, Dst Port: 80, Seq: 1, Ack: 1, Len: 389
▸ Hypertext Transfer Protocol
```

```
GET /wireshark-labs/HTTP-wireshark-file2.html HTTP/1.1
Host: gaia.cs.umass.edu
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:125.0) Gecko/20100101 Firefox/125.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Upgrade-Insecure-Requests: 1
If-Modified-Since: Tue, 07 May 2024 05:59:01 GMT
If-None-Match: "173-617d6e1f6b410"

HTTP/1.1 304 Not Modified
Date: Tue, 07 May 2024 13:46:19 GMT
Server: Apache/2.4.6 (CentOS) OpenSSL/1.0.2k-fips PHP/7.4.33 mod_perl/2.0.11 Perl/v5.16.3
Connection: Keep-Alive
Keep-Alive: timeout=5, max=100
ETag: "173-617d6e1f6b410"
```

```
▸ Frame 305: 541 bytes on wire (4328 bits), 541 bytes captured (4328 bits) on interface enp1s0f0, id 0
▸ Ethernet II, Src: IntelCor_c3:6e:04 (b4:96:91:c3:6e:04), Dst: Cisco_61:8d:5f (5c:5a:c7:61:8d:5f)
▸ Internet Protocol Version 4, Src: 192.168.131.140, Dst: 128.119.245.12
▸ Transmission Control Protocol, Src Port: 34158, Dst Port: 80, Seq: 1, Ack: 1, Len: 475
▸ Hypertext Transfer Protocol
```

http

No.	Time	Source	Destination	Protocol	Length	Info
315	4.968643672	192.168.131.140	128.119.245.12	HTTP	455	GET /wireshark-labs/HTTP-wireshark-file2.html HTTP/1.1
324	5.119786910	128.119.245.12	192.168.131.140	HTTP	796	HTTP/1.1 200 OK (text/html)
326	5.158235925	192.168.131.140	128.119.245.12	HTTP	412	GET /favicon.ico HTTP/1.1
328	5.308629041	128.119.245.12	192.168.131.140	HTTP	550	HTTP/1.1 404 Not Found (text/html)
569	9.772708650	192.168.131.140	128.119.245.12	HTTP	541	GET /wireshark-labs/HTTP-wireshark-file2.html HTTP/1.1
571	9.922733509	128.119.245.12	192.168.131.140	HTTP	305	HTTP/1.1 304 Not Modified

Conteste las siguientes preguntas:

7. Revise el contenido de la primera solicitud HTTP GET. ¿Hay algún encabezado `IF-MODIFIED-SINCE`?

   No

8. Revise el contenido de la primera respuesta del servidor. ¿Qué código de estado retornó el servidor? ¿El servidor envió explícitamente el contenido de la página web? ¿Cómo puede demostrarlo?

   Retornó `HTTP/1.1 200 OK` , y si envió explícitamente el contenido de la página web, se puede demostrar porque se ve en el <html> </html>

```
<html>

Congratulations again!  Now you've downloaded the file lab2-2.html. <br>
This file's last modification date will not change.  <p>
Thus  if you download this multiple times on your browser, a complete copy <br>
will only be sent once by the server due to the inclusion of the IN-MODIFIED-SINCE<br>
field in your browser's HTTP GET request to the server.

</html>
```

9. Ahora revise el contenido de la segunda solicitud HTTP GET. ¿Hay algún encabezado `IF-MODIFIED-SINCE`? Si lo hay, ¿qué información aparece a continuación?

```
If-Modified-Since: Tue, 07 May 2024 05:59:01 GMT
```

10. Revise el contenido de la segunda respuesta del servidor. ¿Qué código de estado retornó el servidor? ¿El servidor envió explícitamente el contenido de la página web? Explique.

El código que retornó fue `HTTP/1.1 304 Not Modified` , y no mandó nuevamente el contenido explícito de la página web, pues probablemente se encuentra en el caché del navegador debido a que ya se había accedido a la página antes.

c. Documentos HTML con objetos embebidos

En esta sección estudiaremos qué ocurre cuando el navegador descarga una página con objetos embebidos, en este caso, imágenes que pueden estar en el mismo servidor, o en otro. Para ello:

- Inicie la captura con Wireshark. Emplee el filtro `http`.
- Acceda a la siguiente página: http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file4.html . Se cargará una página con dos imágenes. La primera (el logo de Pearson) reside en `gaia.cs.umass.edu`. La segunda (la tapa de un libro) reside en `kurose.cslash.net`.
- Una vez que las imágenes carguen, detenga la captura en Wireshark.

http

No.	Time	Source	Destination	Protocol	Length	Info
222	3.279199530	192.168.131.140	128.119.245.12	HTTP	455	GET /wireshark-labs/HTTP-wireshark-file4.html HTTP/1.1
226	3.436881971	128.119.245.12	192.168.131.140	HTTP	1367	HTTP/1.1 200 OK (text/html)
228	3.462681328	192.168.131.140	128.119.245.12	HTTP	412	GET /pearson.png HTTP/1.1
239	3.466060277	192.168.131.140	178.79.137.164	HTTP	379	GET /8E_cover_small.jpg HTTP/1.1
244	3.617252692	128.119.245.12	192.168.131.140	HTTP	781	HTTP/1.1 200 OK (PNG)
246	3.810419137	178.79.137.164	192.168.131.140	HTTP	237	HTTP/1.1 301 Moved Permanently
282	4.174672441	192.168.131.140	2.21.75.180	OCSP	489	Request
284	4.250095773	2.21.75.180	192.168.131.140	OCSP	954	Response

```
GET /wireshark-labs/HTTP-wireshark-file4.html HTTP/1.1
Host: gaia.cs.umass.edu
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:125.0) Gecko/20100101 Firefox/125.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Upgrade-Insecure-Requests: 1

HTTP/1.1 200 OK
Date: Tue, 07 May 2024 13:54:20 GMT
Server: Apache/2.4.6 (CentOS) OpenSSL/1.0.2k-fips PHP/7.4.33 mod_perl/2.0.11 Perl/v5.16.3
Last-Modified: Tue, 07 May 2024 05:59:01 GMT
ETag: "3ae-617d6e1f6ac40"
Accept-Ranges: bytes
Content-Length: 942
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=UTF-8

<html>
<head>
<title>Lab2-4 file: Embedded URLs</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>

<body bgcolor="#FFFFFF" text="#000000">

<p>
<img src="http://gaia.cs.umass.edu/pearson.png" WIDTH="140" HEIGHT="82" > </p>
<p>This little HTML file is being served by gaia.cs.umass.edu.
It contains two embedded images. The image above, also served from the
gaia.cs.umass.edu web site, is the logo of our publisher, Pearson.
The image of our 8th edition book cover below is stored at, and served from,
  a  WWW server kurose.cslash.net in France:</p>
<p align="left"><img src="http://kurose.cslash.net/8E_cover_small.jpg"
                     width="168" height="220"></p>
And while we have your attention, you might want to take time to check out the
                     available open resources for this book at
                     <a href="http://gaia.cs.umass.edu/kurose_ross"> http://gaia.cs.umass.edu/kurose_ross</a>.

</body>
</html>
```

```
▸ Frame 222: 455 bytes on wire (3640 bits), 455 bytes captured (3640 bits) on interface enp1s0f0, id 0
▸ Ethernet II, Src: IntelCor_c3:6e:04 (b4:96:91:c3:6e:04), Dst: Cisco_61:8d:5f (5c:5a:c7:61:8d:5f)
▸ Internet Protocol Version 4, Src: 192.168.131.140, Dst: 128.119.245.12
▸ Transmission Control Protocol, Src Port: 56644, Dst Port: 80, Seq: 1, Ack: 1, Len: 389
▸ Hypertext Transfer Protocol
```

Conteste las siguientes preguntas:

11. ¿Cuántas solicitudes GET envió su navegador? ¿A cuáles direcciones IP fueron enviadas las solicitudes GET?

Envío 3 solicitudes:

- 1 para la página principal

```
222 3.279199530   192.168.131.140      128.119.245.12       HTTP       455 GET /wireshark-labs/HTTP-wireshark-file4.html HTTP/1.1
```

- 2 para las imágenes

```
228 3.462681328   192.168.131.140      128.119.245.12       HTTP       412 GET /pearson.png HTTP/1.1

239 3.466060277   192.168.131.140      178.79.137.164       HTTP       379 GET /8E_cover_small.jpg HTTP/1.1
```

12. ¿Podría decir si las imágenes fueron descargadas en serie, o en paralelo? Explique.

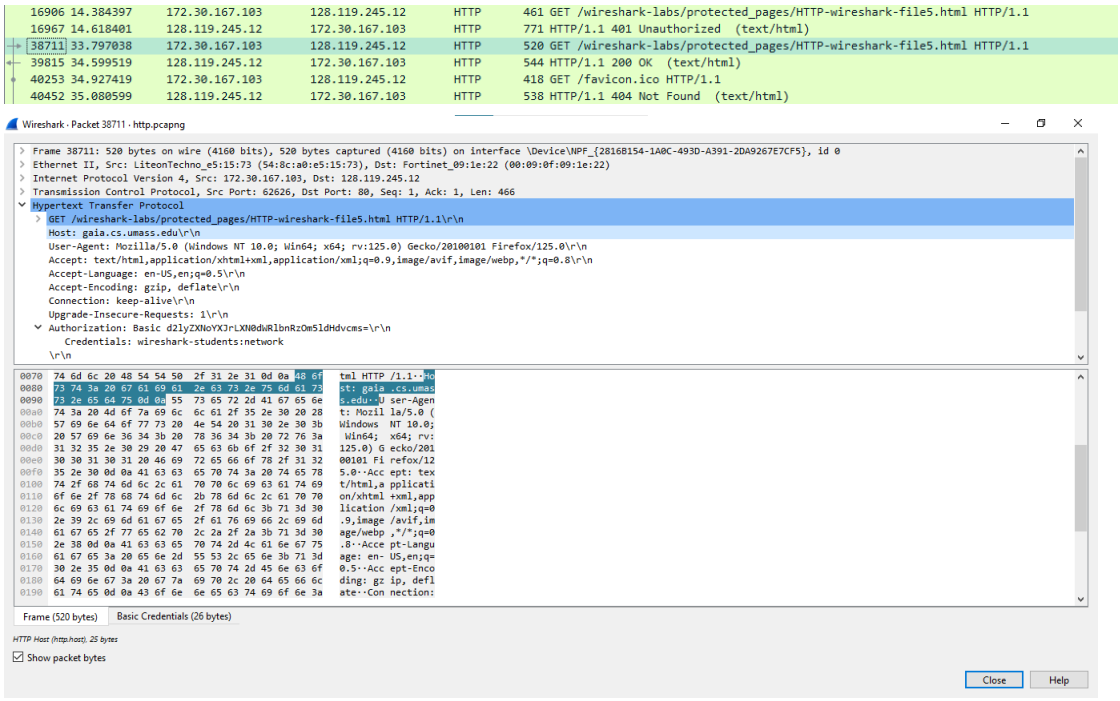
Dado que sabemos que el servidor emplea HTTP/ 1.1, podemos afirmar que las imágenes fueron descargadas en serie, pues HTTP/1.1 descarga imágenes de forma secuencial, no en paralelo. Esto significa que cuando un navegador solicita varias imágenes al mismo tiempo, el servidor HTTP/1.1 responderá a cada solicitud de manera secuencial, una después de otra.

d. Autenticación HTTP

Finalmente se estudiará el proceso de autenticación básica de HTTP, que permite crear sitios web protegidos con usuario y contraseña de una manera muy sencilla y a la vez insegura. Para ello:

- Inicie la captura con Wireshark. Emplee el filtro `http`
- Navegue a la siguiente dirección: http://gaia.cs.umass.edu/wireshark-labs/protected_pages/HTTP-wireshark-file5.html . Cuando se le solicite, introduzca `wireshark-students` como usuario, y `network` como contraseña. Si todo sale bien, aparecerá una página web de dos líneas.

- Detenga la captura en Wireshark. Se podrán observar dos solicitudes GET.



```
GET /wireshark-labs/protected_pages/HTTP-wireshark-file5.html HTTP/1.1
Host: gaia.cs.umass.edu
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:125.0) Gecko/20100101 Firefox/125.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Authorization: Basic d2lyZXNoYXJrLXN0dWRlbnRzOm5ldHdvcms=

HTTP/1.1 200 OK
Date: Fri, 10 May 2024 18:26:59 GMT
Server: Apache/2.4.6 (CentOS) OpenSSL/1.0.2k-fips PHP/7.4.33 mod_perl/2.0.11 Perl/v5.16.3
Last-Modified: Fri, 10 May 2024 05:59:01 GMT
ETag: "84-618133b79eeed"
Accept-Ranges: bytes
Content-Length: 132
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=UTF-8


<html>

This page is password protected!  If you're seeing this, you've downloaded the page correctly <br>
Congratulations!
</html>GET /favicon.ico HTTP/1.1
Host: gaia.cs.umass.edu
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:125.0) Gecko/20100101 Firefox/125.0
Accept: image/avif,image/webp,*/*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Referer: http://gaia.cs.umass.edu/wireshark-labs/protected_pages/HTTP-wireshark-file5.html
```

Conteste las siguientes preguntas:

13. ¿Cuál es el código de respuesta a la primera solicitud GET de su navegador?

    GET /wireshark-labs/protected_pages/HTTP-wireshark-file5.html HTTP/1.1
    Host: gaia.cs.umass.edu
    User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:125.0) Gecko/20100101 Firefox/125.0
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
    Accept-Language: en-US,en;q=0.5
    Accept-Encoding: gzip, deflate
    Connection: keep-alive
    Upgrade-Insecure-Requests: 1
    HTTP/1.1 401 Unauthorized
    Date: Fri, 10 May 2024 18:26:39 GMT
    Server: Apache/2.4.6 (CentOS) OpenSSL/1.0.2k-fips PHP/7.4.33 mod_perl/2.0.11 Perl/v5.16.3
    WWW-Authenticate: Basic realm="wireshark-students only"
    Content-Length: 381
    Keep-Alive: timeout=5, max=100
    Connection: Keep-Alive
    Content-Type: text/html; charset=iso-8859-1

    Lanza el mensaje en azul.

14. Cuando su navegador envía el GET por segunda vez, ¿qué encabezado adicional se incluye con la solicitud?

```
GET /wireshark-labs/protected_pages/HTTP-wireshark-file5.html HTTP/1.1
Host: gaia.cs.umass.edu
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:125.0) Gecko/20100101 Firefox/125.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Authorization: Basic d2lyZXNoYXJrLXN0dWRlbnRzOm5ldHdvcms=
```

Vemos que el encabezado adicional es el de “Authorization” con el valor de “Basic d2lyZXNoYXJrLXN0dWRlbnRzOm5ldHdvcms=”

15. Decodifique la cadena que aparece a continuación del encabezado nuevo empleando un decodificador Base64 (puede buscar uno en Google). ¿Qué cadena de texto aparece al decodificar la cadena original?

    Al decodificar la cadena original, se obtiene “wireshark-students:network”