Práctica 2: HTTP

Después de mojarnos los pies con el detector de paquetes Wireshark en el laboratorio introductorio, estamos listos para usar Wireshark para investigar protocolos en operación!! En este laboratorio, lo haremos explorarando varios aspectos del protocolo HTTP: la interacción GET / respuesta básica, formatos de mensajes HTTP, recuperación de archivos HTML grandes, recuperación de archivos HTML con objetos, y autenticación y seguridad HTTP.

Antes de comenzar estos laboratorios, es recomendable leer la Sección 2.2 del libro de texto.

Parte 1: Interacción básica del comando GET de HTTP

Comencemos nuestra exploración de HTTP descargando un archivo HTML muy simple, uno que es muy corto y no contiene objetos incrustados. Haz lo siguiente:

1. Inicia tu navegador web.
2. Inicia el sniffer de paquetes Wireshark, como se describe en el laboratorio de introducción (pero aún no comiences a capturar paquetes). Ingrese "http" (solo las letras, no la cita) en la ventana de especificación de filtro de visualización, de modo que solo HTTP capturado los mensajes se mostrarán más adelante en la ventana de listado de paquetes. (Estamos interesados sólo en el protocolo HTTP, y así no verás el desorden de todo paquetes capturados).
3. Espera un poco más de un minuto (veremos por qué en breve), y luego comienza la ***Captura de paquetes Wireshark***.
4. Escribe lo siguiente en el navegador http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file1.html (El navegador debe mostrar un archivomuy simple de HTML de una sola línea)
5. Deten la captura de paquetes de Wireshark.

Tu ventana de Wireshark se debe parecer a la imagen de la Figura 1. Si no tienes forma de tomar una captura de Wireshark desde tu red (pueden haber restricciones en el cortafuegos, estar detrás de un servidor NAT, entre otros) puedes descargarla en el siguiente enlace: http://gaia.cs.umass.edu/wireshark-labs/wireshark-traces.zip, extrae el archivo, puedes cargarlo en Wireshark y ver el trazado usando el menú desplegable Archivo, seleccionando Abrir, y luego seleccionando el archivo de rastreo http-ethereal-trace-1. La pantalla resultante debería verse similar a la Figura 1. Pantalla de Wireshark después de http://gaia.cs.umass.edu/wireshark-labs/ HTTPwireshark-archivo1.html sea descargado por tu navegador. Las trazas en este archivo zip fueron recolectadas por Wireshark corriendo en una de las computadoras, mientras se realizan los pasos indicados en el laboratorio de Wireshark.

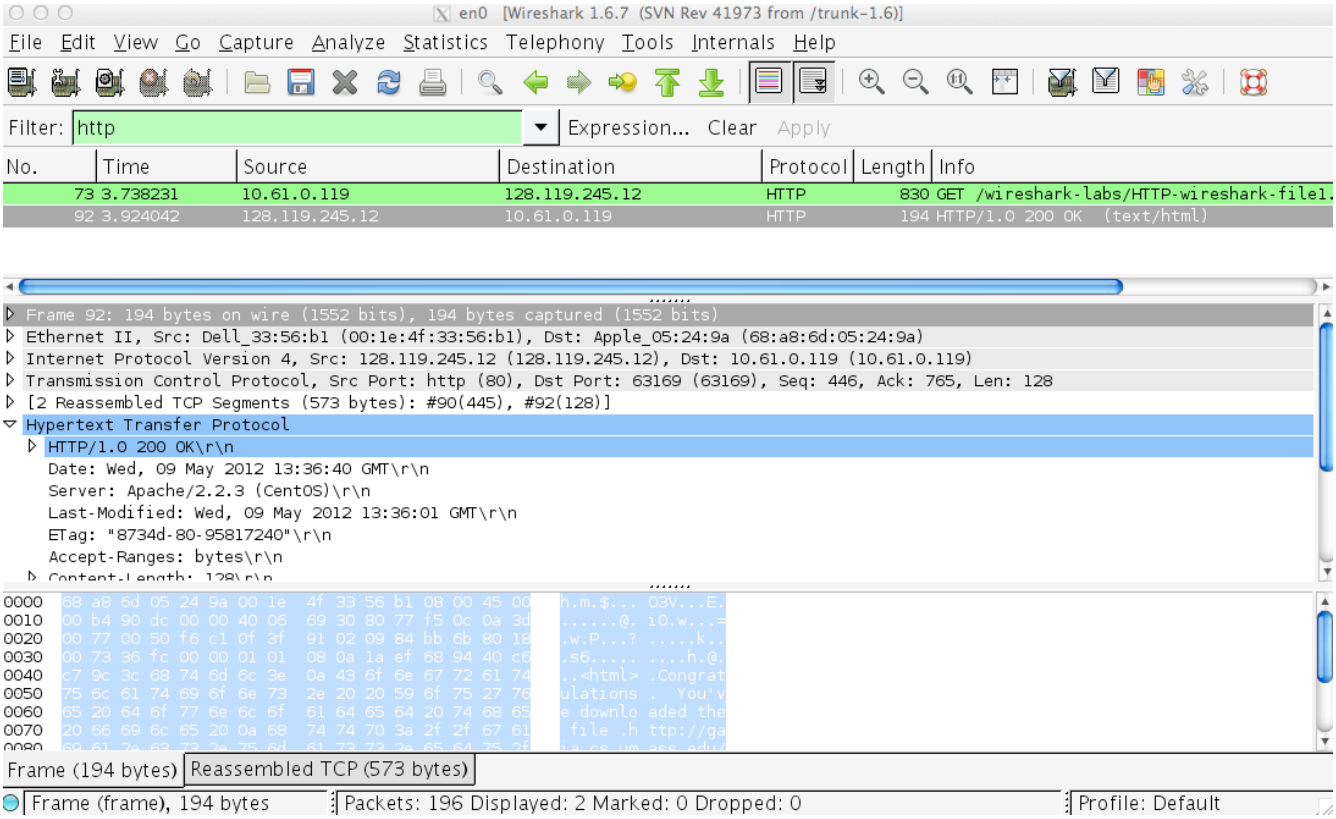

Figura 1: Lo que muestra Wireshark después de que ha sido descargado por el navegador el archivo en el enlace: http://gaia.cs.umass.edu/wireshark-labs/ HTTPwireshark-file1.html

El ejemplo en la Figura 1 muestra en la ventana el listado de paquetes de dos mensajes HTTP que fueron capturados: el mensaje GET (desde el navegador al servidor web gaia.cs.umass.edu) y el mensaje de respuesta del servidor a su navegador. El contenido del paquete en la ventana muestra los detalles del mensaje seleccionado (en este caso, el mensaje HTTP OK, que se resalta en la ventana de listado de paquetes). Recordemos que desde el mensaje HTTP fue llevado dentro de un segmento TCP, que fue llevado dentro de un datagrama IP, que era llevado dentro de un marco Ethernet, Wireshark muestra el Marco, Ethernet, IP y TCP y también la información del paquete. Queremos minimizar la cantidad de datos que no son HTTP (por el momento sólo vamos a revisar lo que sucede con el protocolo HTTP), así que asegurate de que los cuadros en el extremo izquierdo de la trama, Ethernet, IP y TCP la información tiene un signo más o un triángulo que apunta hacia la derecha (lo que significa esté oculto), y la línea HTTP tenga un signo menos o un signo hacia abajo triángulo (lo que significa que se muestra toda la información sobre el mensaje HTTP).

Al mirar la información en el GET del HTTP y los mensajes de respuesta, responde las siguientes preguntas. Al responder las siguientes preguntas, debe imprimir los mensajes GET y de respuesta e indicar en qué parte del mensaje encontró la información que responde las siguientes preguntas. Cuando entregues la tarea, anota la salida para que quede claro en qué parte del resultado está obteniendo la información para su respuesta (resalta el resultado con otro color).

1. ¿Está el navegador ejecutando HTTP versión 1.0 o 1.1? ¿Qué versión de HTTP tiene el servidor en funcionamiento?
2. ¿Qué idiomas (si corresponde) indica tu navegador que puede aceptar el servidor?
3. ¿Cuál es la dirección IP de tu computadora? ¿Cuál es la dirección IP del servidor gaia.cs.umass.edu?

4. ¿Cuál es el código de estado devuelto por el servidor al navegador?
5. ¿Cuándo fue modificado por última vez el archivo HTML que obtuviste del servidor?
6. ¿Cuántos bytes de contenido se están devolviendo a tu navegador?
7. Al inspeccionar los datos brutos en la ventana de contenido del paquete, ¿ves algún encabezado dentro de los datos que no se muestre en la ventana de paquetes? Si es así, nombra uno.

En su respuesta a la pregunta 5, es posible que te haya sorprendido descubrir que el documento que acabas de descargar se modificó por última vez en un minuto antes de descargar el documento. Eso es porque (para este archivo en particular), el servidor gaia.cs.umass.edu está configurando la hora del último cambio del archivo para que sea la hora actual, y lo hace una vez por minuto. Por lo tanto, si esperas un minuto entre accesos, el archivo parecerá haber sido modificado recientemente y, por lo tanto, su navegador descargará una copia "nueva" del documento.

2. La interacción de respuesta del GET condicional de HTTP

Recuerde de la Sección 2.2.5 del texto, que la mayoría de los navegadores web realizan caché de objetos y, por lo tanto, realizan un GET condicional al recuperar un objeto HTTP. Antes de realizar los siguientes pasos, asegúrate de que la memoria caché del navegador esté vacía. (Para hacer esto en Firefox, seleccione Herramientas-> Borrar historial reciente y marque la casilla Caché, o para Internet Explorer, seleccione Herramientas-> Opciones de Internet-> Eliminar archivo; estas acciones eliminarán los archivos almacenados en caché de la memoria caché de su navegador) Ahora realiza los siguientes pasos:

• Inicia el sniffer de paquetes Wireshark
• Ingresa la siguiente URL en tu navegador: http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file2.html (El navegador debe mostrar un archivo HTML de cinco líneas muy simple)
• Ingresa rápidamente la misma URL en su navegador nuevamente (o simplemente seleccione el botón de actualización en su navegador)
• Detén la captura de paquetes de Wireshark e ingresa "http" en la ventana de especificación del filtro de visualización, de modo que solo se muestren los mensajes HTTP capturados más adelante en la ventana de listado de paquetes.

(Nota: si no puede ejecutar Wireshark en una conexión de red en vivo, puede usar la traza de paquetes http-ethereal-trace-2 para responder las siguientes preguntas; la misma está en el paquete .zip que se indicó en la sección anterior. Este archivo de rastreo se recopiló al realizar los pasos anteriores en una de las computadoras del autor).

Responde las siguientes preguntas:

8. Inspecciona el contenido de la primera solicitud HTTP GET de tu navegador al servidor. ¿Ves una línea "IF-MODIFIED-SINCE" en el GET de HTTP?
9. Inspecciona el contenido de la respuesta del servidor. ¿El servidor devolvió explícitamente el contenido del archivo? ¿Cómo puedes afirmarlo?
10. Ahora inspecciona el contenido de la segunda solicitud HTTP GET desde tu navegador al servidor. ¿Ves una línea "IF-MODIFIED-SINCE:" en HTTP GET? De ser así, ¿qué información sigue al encabezado "IF-MODIFIED-SINCE:"?
11. ¿Cuál es el código de estado HTTP y la frase devuelta por el servidor en respuesta a este segundo HTTP GET? ¿El servidor devolvió explícitamente el contenido del archivo? Explica.

3. Descargando documentos largos

En nuestros ejemplos hasta ahora, los documentos descargados han sido archivos HTML simples y cortos. Veamos a continuación qué sucede cuando descargamos un archivo HTML largo. Haz lo siguiente:

• Inicia tu navegador web y asegúrate de que la memoria caché de tu navegador esté desactivada.
• Inicia el sniffer de paquetes Wireshark
• Ingresa la siguiente URL en el navegador http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file3.html
Tu navegador debe mostrar la Declaración de Derechos de EE. UU. Es bastante larga.
• Detén la captura de paquetes Wireshark e ingresa "http" en la especificación del filtro de visualización, para que sólo se muestren los mensajes HTTP capturados.

(Nota: si no puede ejecutar Wireshark en una conexión de red en vivo, puede utilice la traza del paquete http-ethereal-trace-3 para responder a las siguientes preguntas; ver nota al pie 1. Este archivo de rastreo se recopiló al realizar los pasos anteriores en uno de las computadoras del autor.)

En la ventana de listado de paquetes, deberías ver el mensaje HTTP GET, seguido de una respuesta TCP de paquetes múltiples a su solicitud HTTP GET. Este paquete múltiple merece una explicación. Recuerda de la Sección 2.2 (ver Figura 2.9 en el texto) que el mensaje de respuesta HTTP consiste en una línea de estado, seguida de líneas de encabezado, seguido de una línea en blanco, seguido por el cuerpo de la entidad. En el caso de nuestro HTTP GET, El cuerpo de la entidad en la respuesta es todo el archivo HTML solicitado. En nuestro caso aquí, El archivo HTML es bastante largo, y en 4500 bytes es demasiado grande para caber en un paquete TCP. Por lo que un solo mensaje de respuesta HTTP se divide en varias partes, en cada pieza está contenida en un segmento separado. En Las versiones más recientes de Wireshark, se indica que cada segmento del paquete, y el hecho de que la única respuesta HTTP estaba fragmentada a través de múltiples paquetes TCP se puede ver mediante el "segmento TCP de una PDU reensamblada" en la columna Información de la pantalla de Wireshark. Las versiones anteriores de Wireshark usaban la frase "Continuación" para indicar que todo el contenido de un mensaje HTTP se rompió en varios TCP segmentos ... Hacemos hincapié aquí que no hay mensaje de "Continuación" en HTTP!

Responde las siguientes preguntas:

12. ¿Cuántos mensajes de solicitud HTTP GET envió su navegador?
13. ¿Qué número de paquete en la traza contiene el código de estado y la frase asociada con la respuesta a la solicitud HTTP GET?
14. ¿Cuál es el código de estado y la frase en la respuesta?
15. ¿Cuántos segmentos de TCP que contenían datos se necesitaban para transportar HTTP único?

4. Documentos HTML con Objects

Ahora que hemos visto cómo Wireshark muestra el tráfico de paquetes capturados para HTML de grandes archivos, podemos ver lo que sucede cuando su navegador descarga un archivo con objetos incrustados objetos, es decir, un archivo que incluye otros objetos (en el ejemplo a continuación, archivos de imágenes) que son almacenados en otro servidor.

Haz lo siguiente:
• Inicia tu navegador web y asegúrate de que la memoria caché de su navegador esté desactivada, discutido arriba.
• Inicia el sniffer de paquetes Wireshark
• Ingresa la siguiente URL: http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file4.html en tu navegador.

El navegador debe mostrar un archivo HTML corto con dos imágenes. Estas dos imágenes se referencian en el archivo HTML base. Es decir, las imágenes en sí mismas no son contenido en el HTML; en su lugar, las URL de las imágenes están contenidas en archivo HTML descargado. Como se discutió en clase, el navegador tendrá que recuperar estos logotipos de los sitios web indicados. El logotipo de nuestra editorial alojado en el sitio web gaia.cs.umass.edu y la imagen de la portada de nuestro logo de la quinta edición (una de nuestras portadas favoritas) que se almacena en el servidor caite.cs.umass.edu. (Estos son dos servidores web diferentes dentro de cs.umass.edu).

• Detén la captura de paquetes Wireshark e ingresa "http" en la especificación del filtro de visualización, para que solo se muestren los mensajes HTTP capturados.

(Nota: si no puede ejecutar Wireshark en una conexión de red en vivo, puede utiliza el rastreo de paquetes http-ethereal-trace-4 para responder a las siguientes preguntas; Este archivo de rastreo se recopiló al realizar los pasos anteriores en uno de las computadoras del autor.)

Responde las siguientes preguntas:
16. ¿Cuántos mensajes de solicitud HTTP GET envió tu navegador? ¿A que direcciones de IP fueron enviadas estas solicitudes GET?
17. ¿Puedes decir si tu navegador descargó las dos imágenes en serie, o si fueron descargadas de los dos sitios web en paralelo? Explique

5 Autenticación HTTP

Finalmente, intentemos visitar un sitio web que esté protegido por contraseña y examinemos la secuencia del mensaje HTTP intercambiado para dicho sitio. La URL http://gaia.cs.umass.edu/wireshark-labs/protected_pages/HTTP-wireshark-file5.html. El nombre de usuario es "wireshark-students" (sin las comillas), y el la contraseña es "red" (nuevamente, sin las comillas). Así que accedamos a este sitio "seguro" protegido por contraseña. Haz lo siguiente:

• Asegúrate de que la memoria caché de su navegador esté desactivada, como se indicó anteriormente, y cierra tu navegador Luego, inicia tu navegador
• Inicia el sniffer de paquetes Wireshark
• Ingresa la siguiente URL en tu navegador:
http://gaia.cs.umass.edu/wireshark-labs/protected_pages/HTTP-wiresharkfile5.html
Escribe el nombre de usuario y la contraseña solicitados en el cuadro emergente.
• Detén la captura de paquetes Wireshark e ingresa "http" en la especificación del filtro de visualización, de modo que sólo se muestren los mensajes HTTP.

(Nota: si no puede ejecutar Wireshark en una conexión de red en vivo, puedes utilizar el rastreo de paquetes http-ethereal-trace-5 . Este archivo de rastreo se recopiló al realizar los pasos anteriores en uno de las computadoras del autor.)

Ahora examinemos la salida de Wireshark. Responde las siguientes preguntas:

18. ¿Cuál es la respuesta del servidor (código de estado y frase) en respuesta a la solicitud inicial que recibió de tu navegador?
19. Cuando tu navegador envía el mensaje HTTP GET por segunda vez, ¿qué campo nuevo está incluido en el mensaje HTTP GET?

El nombre de usuario (wireshark-students) y la contraseña (red) que ingresaste están codificados en la cadena de caracteres (d2lyZXNoYXJrLXN0dWRlbnRzOm5ldHdvcms =) siguiente el encabezado "Autorización: Básica" en el mensaje HTTP GET del cliente. A primera vista puede parecer que el nombre de usuario y contraseña están cifrados, pero simplemente están codificados en un formato conocido como formato Base64. ¡El nombre de usuario y la contraseña no están cifrados! Para comprobarlo, ve a http://www.motobit.com/util/base64-decoder-encoder.asp e ingresa la cadena codificada en base64 d2lyZXNoYXJrLXN0dWRlbnRz y presiona decodificación. Voila! ¡deberías ver el nombre de usuario!

Para ver la contraseña, ingresa el resto de la cadena Om5ldHdvcms = y presiona descodificar. Ya que cualquiera puede descargar una herramienta como Wireshark y olfatear paquetes pasando por tu adaptador de red, y cualquiera puede traducir de Base64 a ASCII (¡acaba de hacerlo!), Debes tener en claro que las contraseñas simples en los sitios WWW no son seguros a menos que se tomen medidas adicionales.

Como veremos en el Capítulo 8, hay formas de hacer que el acceso WWW sea más seguro!!! Sin embargo, claramente necesitaremos algo que va más allá de la autenticación HTTP básica.