WiresharkLab\_HTTP

建議:把比例放到150%圖才會清楚

組別:A10

組長:資管二甲 B0444132 張仁樵(做最多事)

組員:資管二甲 B0444101 李積舜

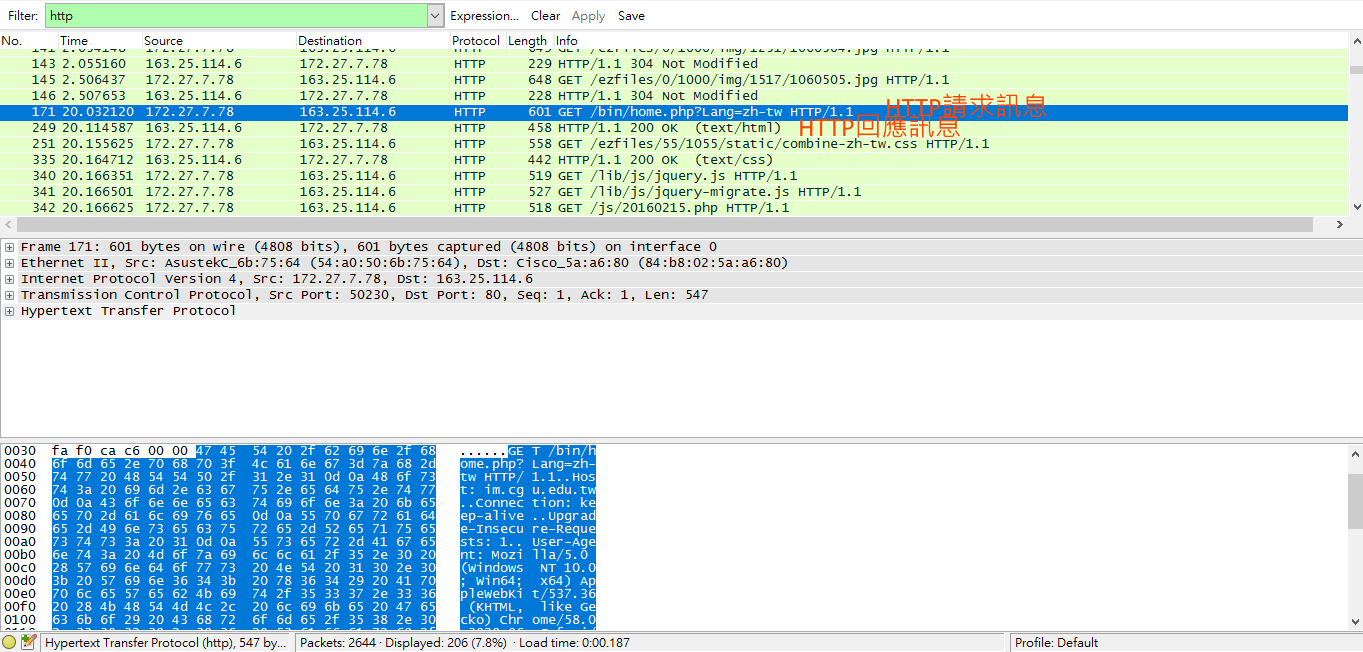
組員:資管二甲 B0444138 潘楚華

組員:資管二甲 B0444146 林子由

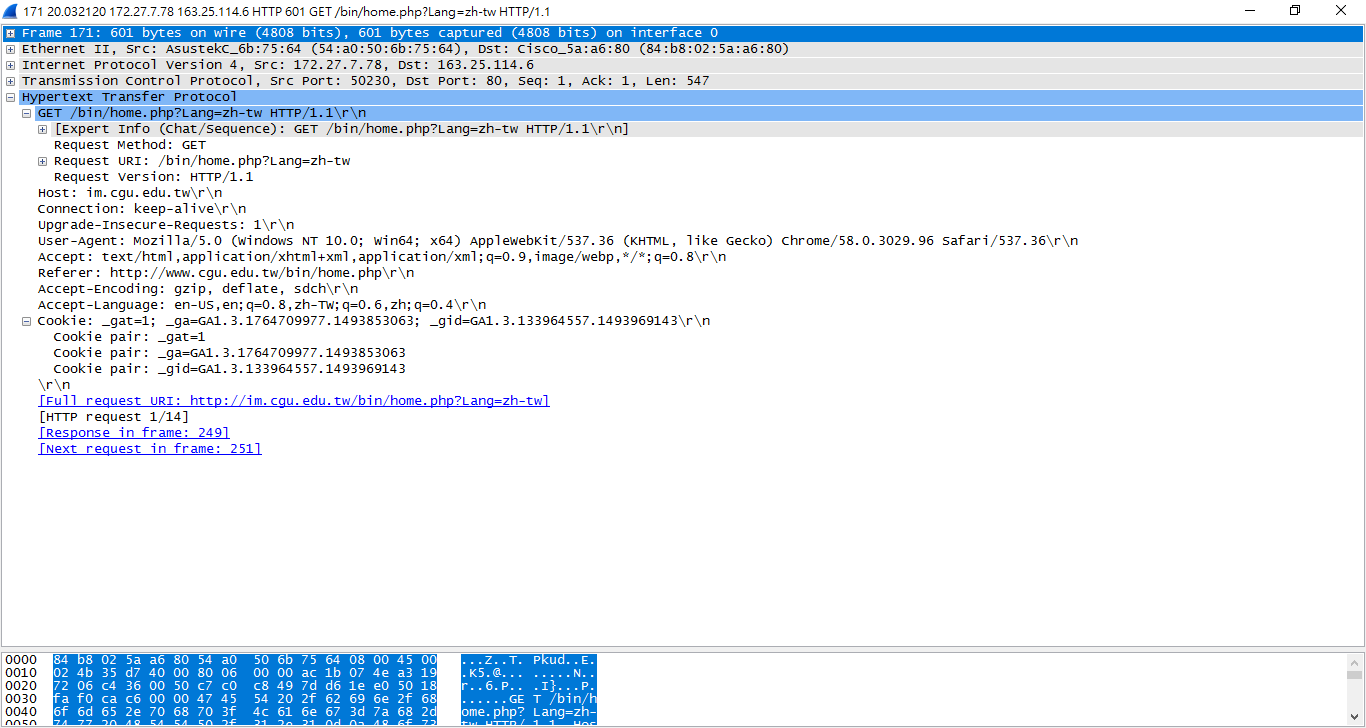
**一、The Basic HTTP GET/response interaction：**



->首先開啟Wireshark準備擷取封包，然後開啟瀏覽器進入系上網站。



->這是擷取後的畫面，NO.171的封包為HTTP請求訊息的封包；NO.249的封包為HTTP回應訊息的封包。



->這是NO.171的封包內容，為HTTP請求訊息。

**GET /bin/home.php?Lang=zh-tw HTTP/1.1** 為請求行，GET為方法，中間則為URL(請求home.php這個物件)，後面HTTP/1.1則為HTTP版本。

1.Host:指出該物件所在的主機。

2.Connection: 針對該連線所預期的選項，keep-alive代表瀏覽器是使用永久性連線。

3. User-Agent：瀏覽器類型，如果Servlet返回的內容與瀏覽器類型有關則該值非常有用。

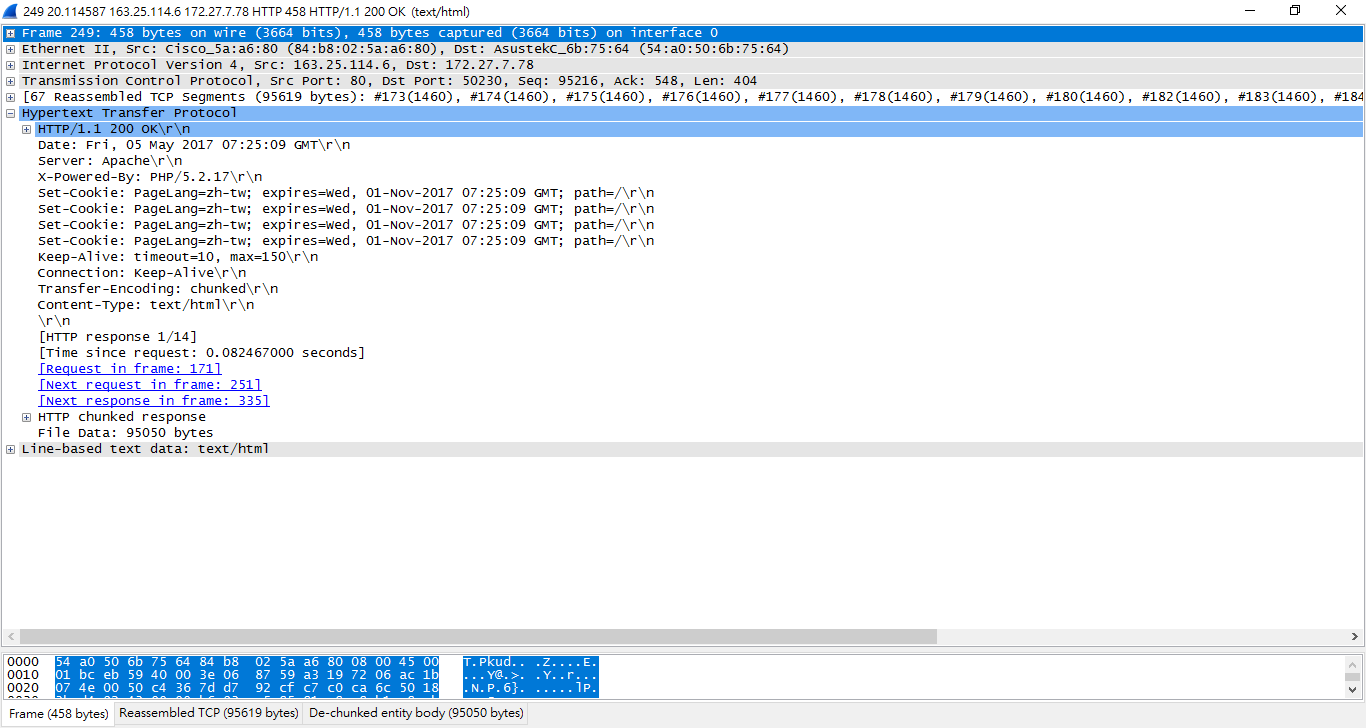
4. Accept:瀏覽器可接受的MIME類型。

5. Referer：包含一個URL，用戶從該URL代表的頁面出發訪問當前請求的頁面。

6. Accept-Encoding：瀏覽器能夠進行解碼的數據編碼方式，比如gzip。Servlet能夠向支持gzip的瀏覽器返回經gzip編碼的HTML頁面。許多情形下這可以減少5到10倍的下載時間。

7. Accept-Language：瀏覽器所希望的語言種類，當服務器能夠提供一種以上的語言版本時要用到。在請求行中有指定為zh-tw(繁體中文)。

8.Cookie: 之前由伺服器通過 Set- Cookie （下文詳述）傳送的一個超文字傳輸協定Cookie。



->這是NO.249的封包內容，為HTTP回應訊息。

**HTTP/1.1 200 OK**  為HTTP狀態行，HTTP/1.1為版本，200為狀態碼，OK為用語。

狀態碼: 1. 200 OK:請求成功，而且相關資訊已經於回應訊息中傳回。

2. 301 Moved Permanently:所請求的物件已經永久移除；新的URL會被標 記於回應訊息的Location:標頭中。用戶端軟體 會自動取得新的URL。

3. 400 Bad Request :這是一個一般性的錯誤碼，表示伺服器無法理解該筆 請求。

4. 404 Not Found:所請求的文件不存在該伺服器上。

5. 505 HTTP Version Not Supported:伺服器不支援所請求的HTTP協定版 本。

1.Date: 此條訊息被傳送時的日期和時間(按照 RFC 7231 中定義的「超文字傳輸協定日期」格式來表示)。

2.Server:則此代表由Apache網頁伺服器所產生。

3.X-Power-By: 表明用於支援當前網頁應用程式的技術（在此為PHP）（版本號細節通常放置在 X-Runtime 或 X-Version 中）。

4. Set-Cookie: HTTP的Cookie

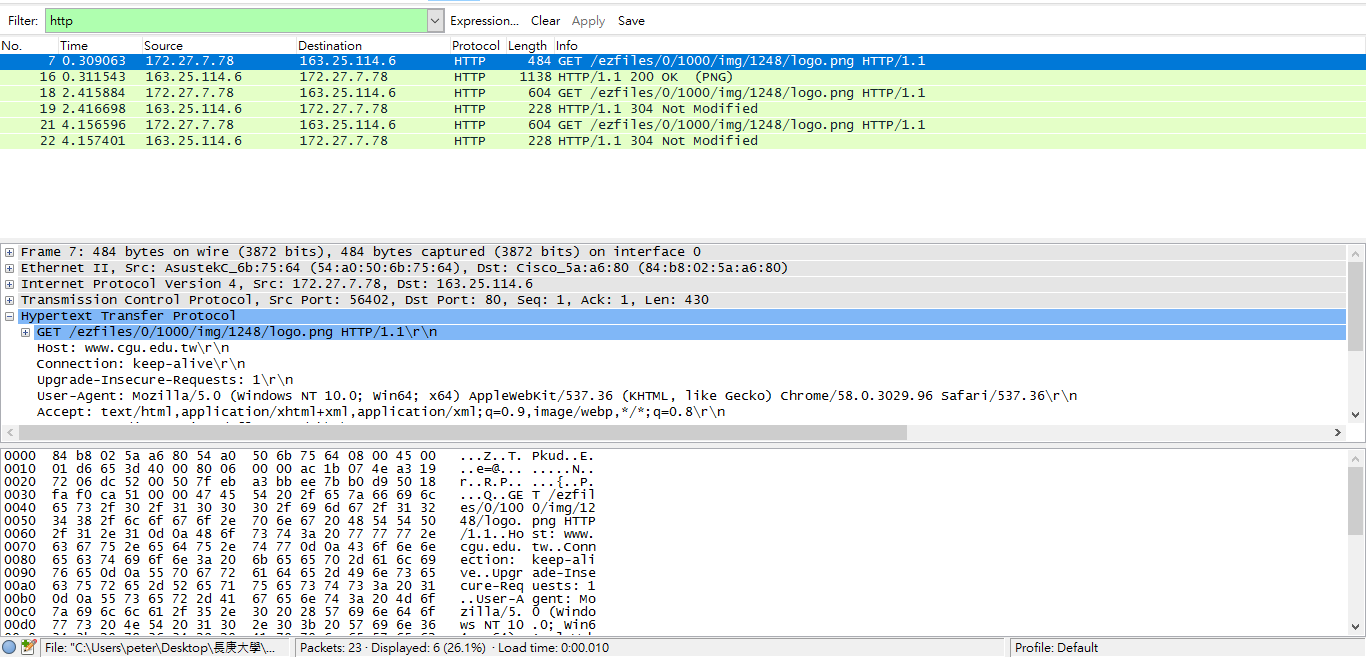
5. Transfer-Encoding: 用來將實體安全地傳輸給用戶的編碼形式。當前定義的方法 包括：分塊(chunked)、壓縮（compress）、縮小（deflate）、壓縮（gzip）、實體（identity）。

6.Content-Type: 當前內容的[MIME](https://zh.wikipedia.org/wiki/MIME)類型(在此是HTML文件)。

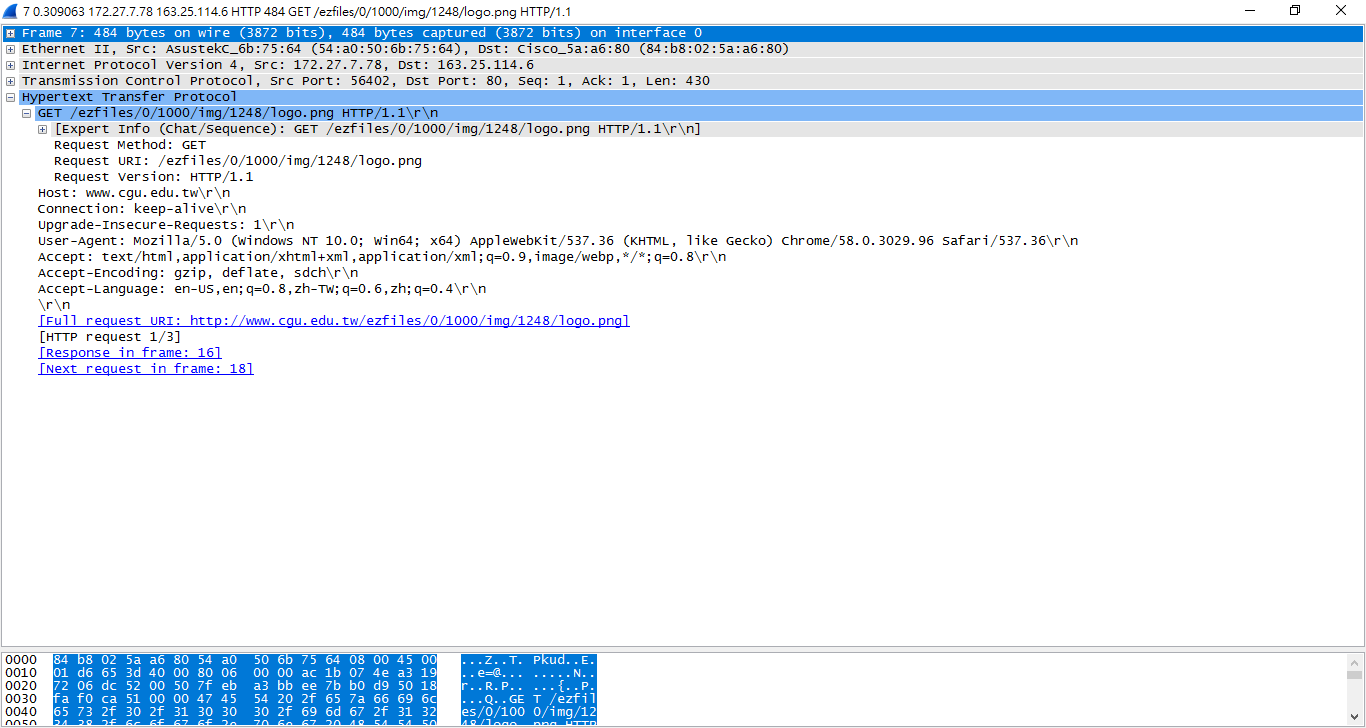
**二、The HTTP CONDITIONAL GET/response interaction：**



->我們利用Cache抓長庚大學首頁的左上角logo.png這個物件(只要先把快取清掉，然後進入一次，並再重新整理一次，就能抓取到利用Cache的封包)。



->NO.7為第一次請求訊息的封包(沒使用Cache)，NO.16為第一次回應訊息的封包(200 OK)，NO.18為第二次請求訊息的封包(有使用Cache)，NO.19為第二次回應訊息的封包(304 Not Modified)，NO.21為第三次請求訊息的封包(有使用Cache)，NO.22為第三次回應訊息的封包(304 Not Modified)。



->NO.7的封包內容，為第一次HTTP請求訊息，並沒有使用Cache。

**GET /ezfiles/0/1000/img/1248/logo.png HTTP/1.1** 為請求行，GET為方法，中間則為URL(請求logo.png這個物件)，後面HTTP/1.1則為HTTP版本。

1.Host:指出該物件所在的主機。

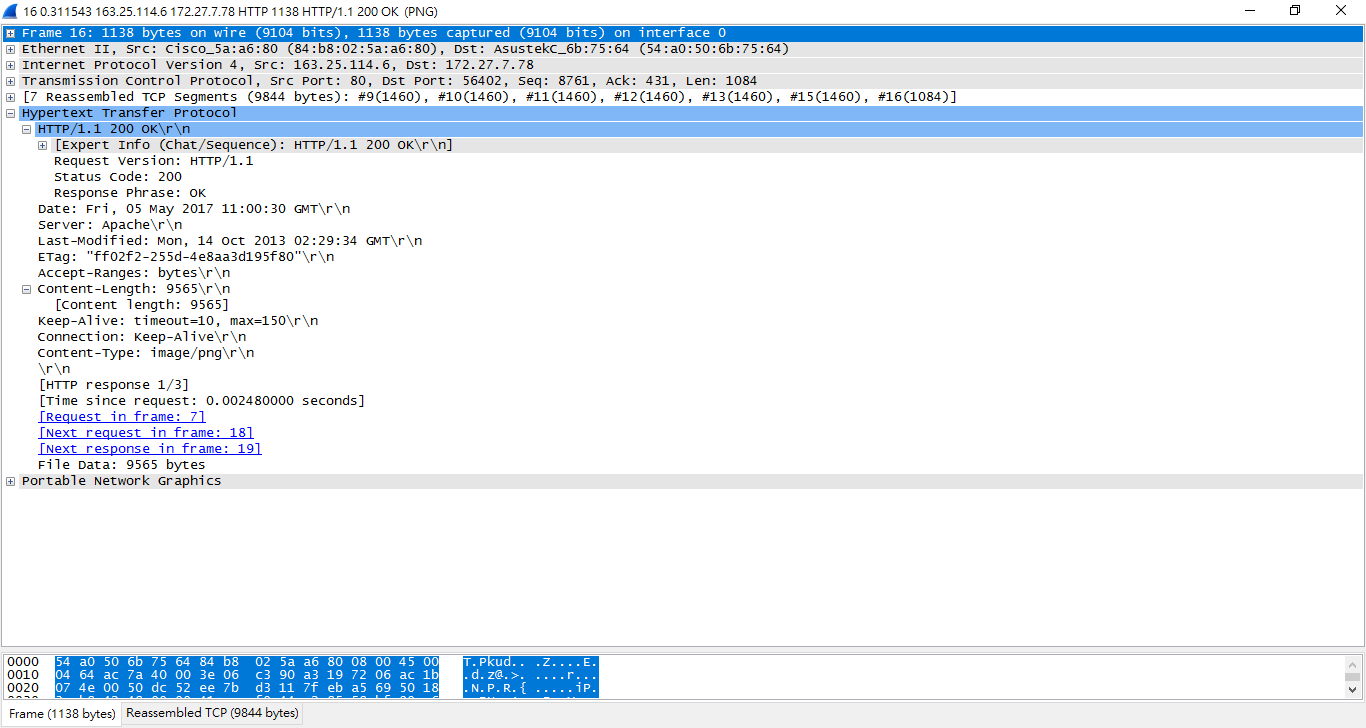
2.Connection: 針對該連線所預期的選項，keep-alive代表瀏覽器是使用永久性連線。

3. User-Agent：瀏覽器類型，如果Servlet返回的內容與瀏覽器類型有關則該值非常有用。

4. Accept:瀏覽器可接受的MIME類型。

5. Accept-Encoding：瀏覽器能夠進行解碼的數據編碼方式，比如gzip。Servlet能夠向支持gzip的瀏覽器返回經gzip編碼的HTML頁面。許多情形下這可以減少5到10倍的下載時間。

6. Accept-Language：瀏覽器所希望的語言種類，當服務器能夠提供一種以上的語言版本時要用到。



->NO.16的封包內容，為第一次HTTP回應訊息，為200 OK。

**HTTP/1.1 200 OK**  為HTTP狀態行，HTTP/1.1為版本，200為狀態碼，OK為用語。

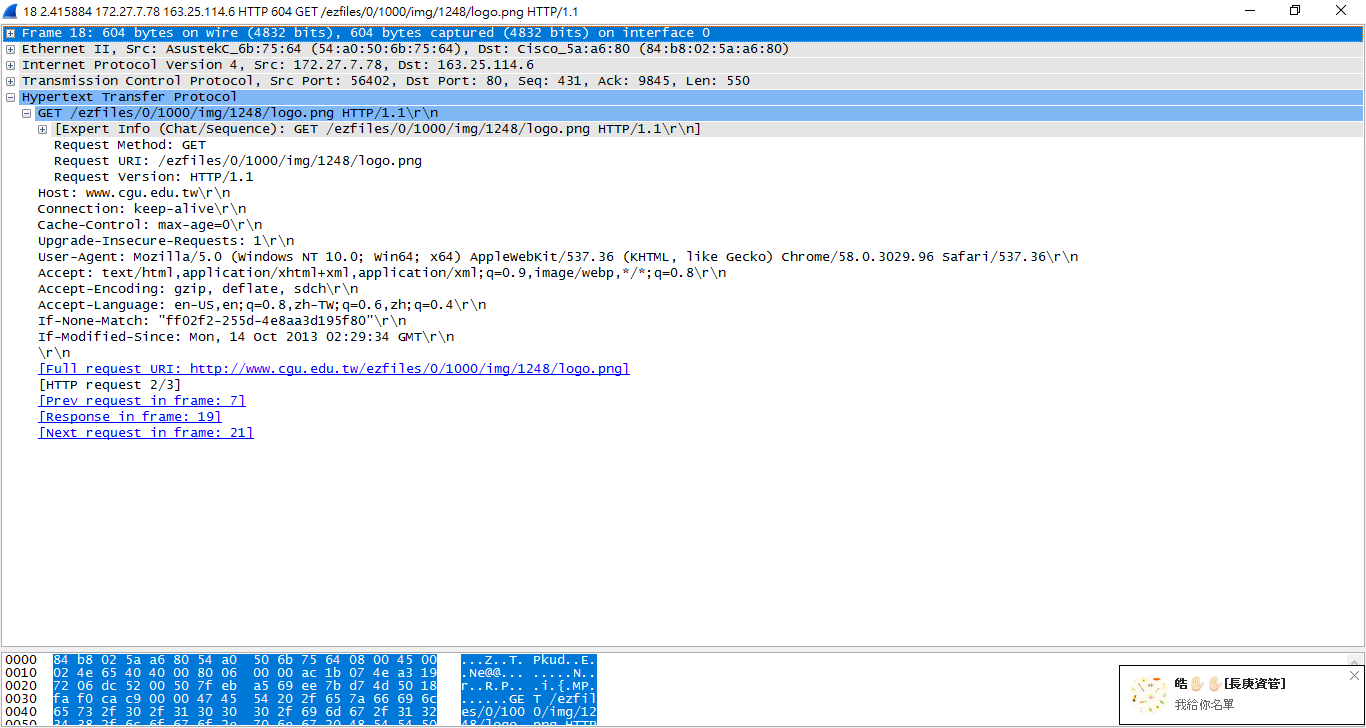
1.Date: 此條訊息被傳送時的日期和時間(按照 RFC 7231 中定義的「超文字傳輸協定日期」格式來表示)。

2.Server:則此代表由Apache網頁伺服器所產生。

3.Last-Modified: 所請求的物件的最後修改日期(按照 RFC 7231 中定義的「超文字傳輸協定日期」格式來表示)。

4.ETag: 對於某個資源的某個特定版本的一個識別元，通常是一個訊息雜湊。

5. Accept-Ranges:這個伺服器支援哪些種類的部分內容範圍。



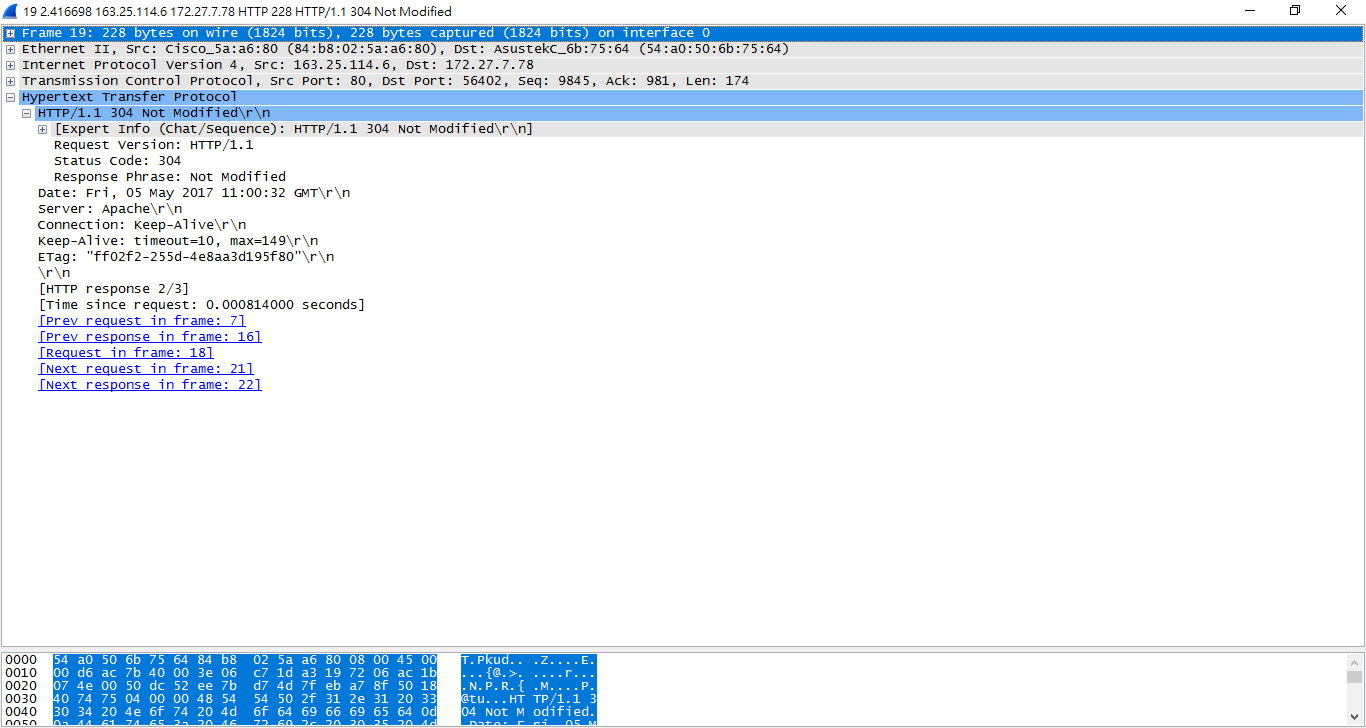
->NO.18的封包內容，為第二次HTTP請求訊息，有使用Cache。

第二次的HTTP請求訊息比第一次多了3個欄位，分別是:

1.Cache-Control: 用來指定在這次的請求/回應鏈中的所有快取機制，都必須遵守的指令

2. If-None-Match: 允許在對應的內容未被修改的情況下返回304未修改（304 Not Modified），參考超文字傳輸協定的[實體標記](https://zh.wikipedia.org/wiki/HTTP_ETag)。

3. If-Modified-Since: 允許在對應的內容未被修改的情況下返回304未修改（304 Not Modified），這個欄位跟回應訊息的Last-Modified數值一樣，這個欄位是在告訴伺服器，只有在指定的日期之後，該物件有被修改過時才需要傳送此物件。



->NO.19的封包內容，為第二次HTTP回應訊息，為304 Not Modified。

**HTTP/1.1 304 Not Modified**  為HTTP狀態行，HTTP/1.1為版本，304為狀態碼，Not Modified為用語。

狀態碼補充: 304 Not Modified

果用戶端傳送了一個帶條件的GET請求且該請求已被允許，而文件的內容（自上次存取以來或者根據請求的條件）並沒有改變，則伺服器應當返回這個狀態碼。304回應禁止包含訊息體，因此始終以訊息頭後的第一個空行結尾。

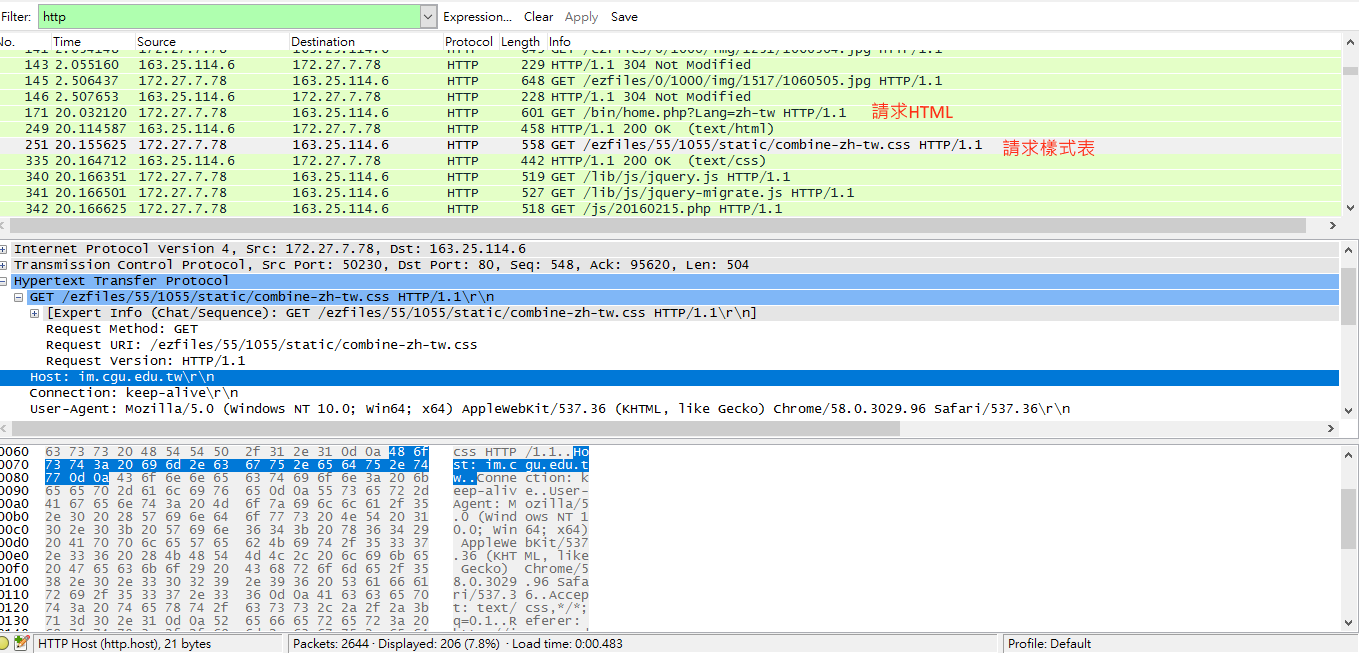
第二次的HTTP回應訊息只比第一次少了1個欄位: Accept-Ranges。

第三次的HTTP請求訊息和回應訊息只是測試多次抓取物件時，是否依然會使用Cache，答案是會的。

**三、Retrieving Long Documents：**

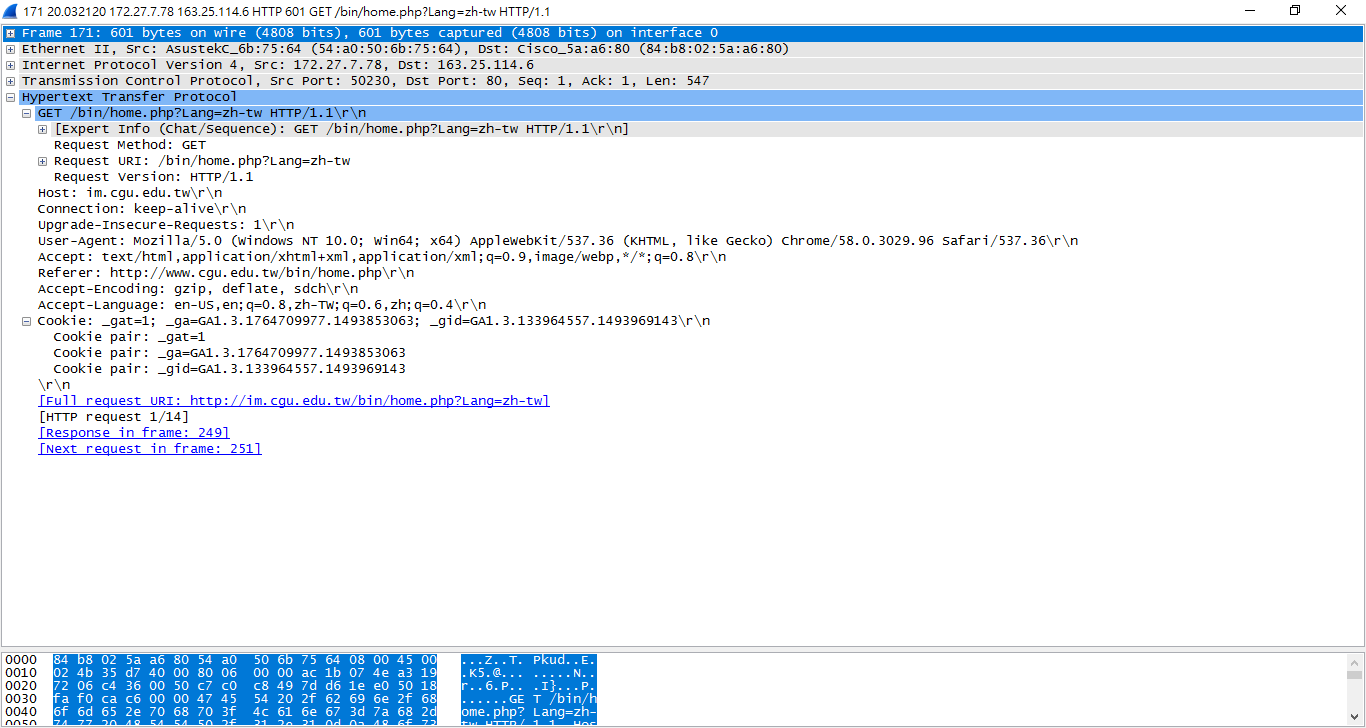


->首先開啟Wireshark準備擷取封包，然後開啟瀏覽器進入系上網站。



­->我們可以看到系上的網站，是有許多回應訊息所組成的，在此我們提HTML和.css(樣式表)來做說明

->NO.171為第一次請求訊息的封包(請求HTML，用PHP寫的)，NO.249為第一次回應訊息的封包(200 OK)，NO.251為第二次請求訊息的封包(請求.css)，NO.335為第二次回應訊息的封包(200 OK)。



->這是NO.171的封包內容，為第一次HTTP請求訊息。

**GET /bin/home.php?Lang=zh-tw HTTP/1.1** 為請求行，GET為方法，中間則為URL(請求home.php這個物件)，後面HTTP/1.1則為HTTP版本。

1.Host:指出該物件所在的主機。

2.Connection: 針對該連線所預期的選項，keep-alive代表瀏覽器是使用永久性連線。

3. User-Agent：瀏覽器類型，如果Servlet返回的內容與瀏覽器類型有關則該值非常有用。

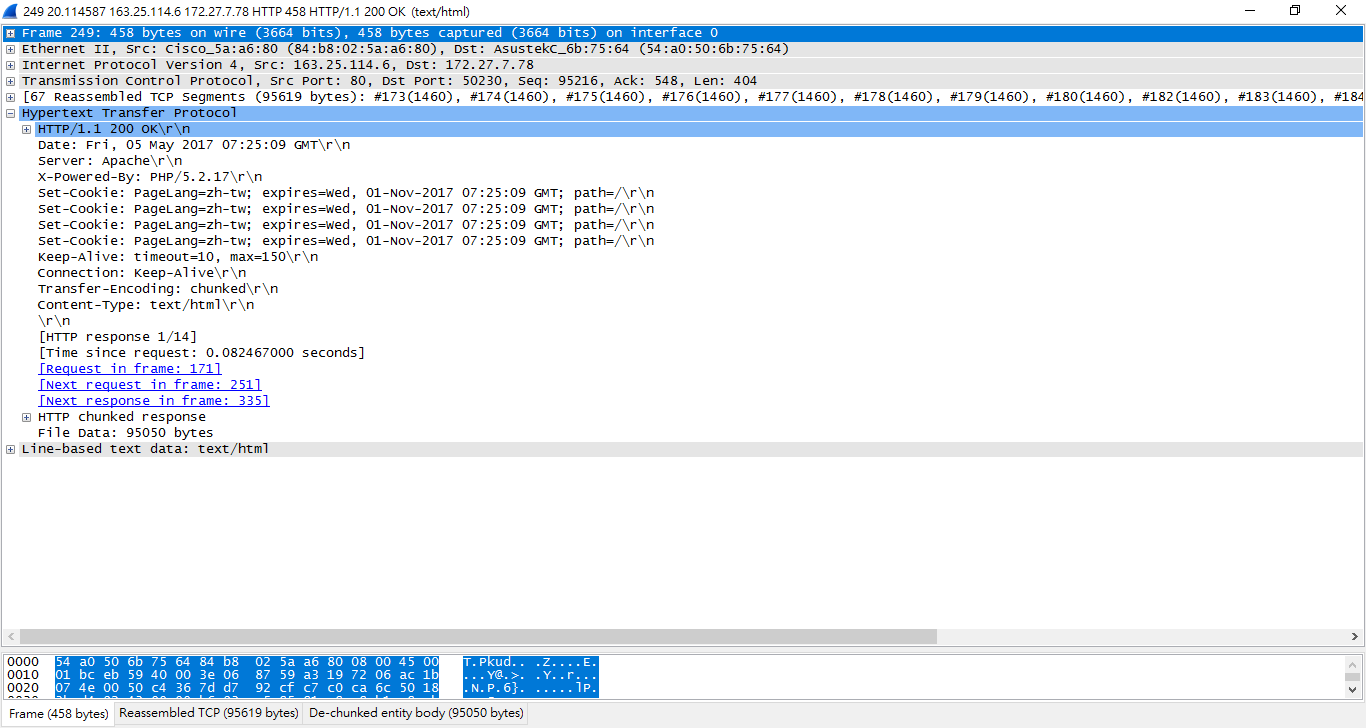
4. Accept:瀏覽器可接受的MIME類型。

5. Referer：包含一個URL，用戶從該URL代表的頁面出發訪問當前請求的頁面。

6. Accept-Encoding：瀏覽器能夠進行解碼的數據編碼方式，比如gzip。Servlet能夠向支持gzip的瀏覽器返回經gzip編碼的HTML頁面。許多情形下這可以減少5到10倍的下載時間。

7. Accept-Language：瀏覽器所希望的語言種類，當服務器能夠提供一種以上的語言版本時要用到。在請求行中有指定為zh-tw(繁體中文)。

8.Cookie: 之前由伺服器通過 Set- Cookie （下文詳述）傳送的一個超文字傳輸協定Cookie。



->這是NO.249的封包內容，為第一次HTTP回應訊息。

**HTTP/1.1 200 OK**  為HTTP狀態行，HTTP/1.1為版本，200為狀態碼，OK為用語。

1.Date: 此條訊息被傳送時的日期和時間(按照 RFC 7231 中定義的「超文字傳輸協定日期」格式來表示)。

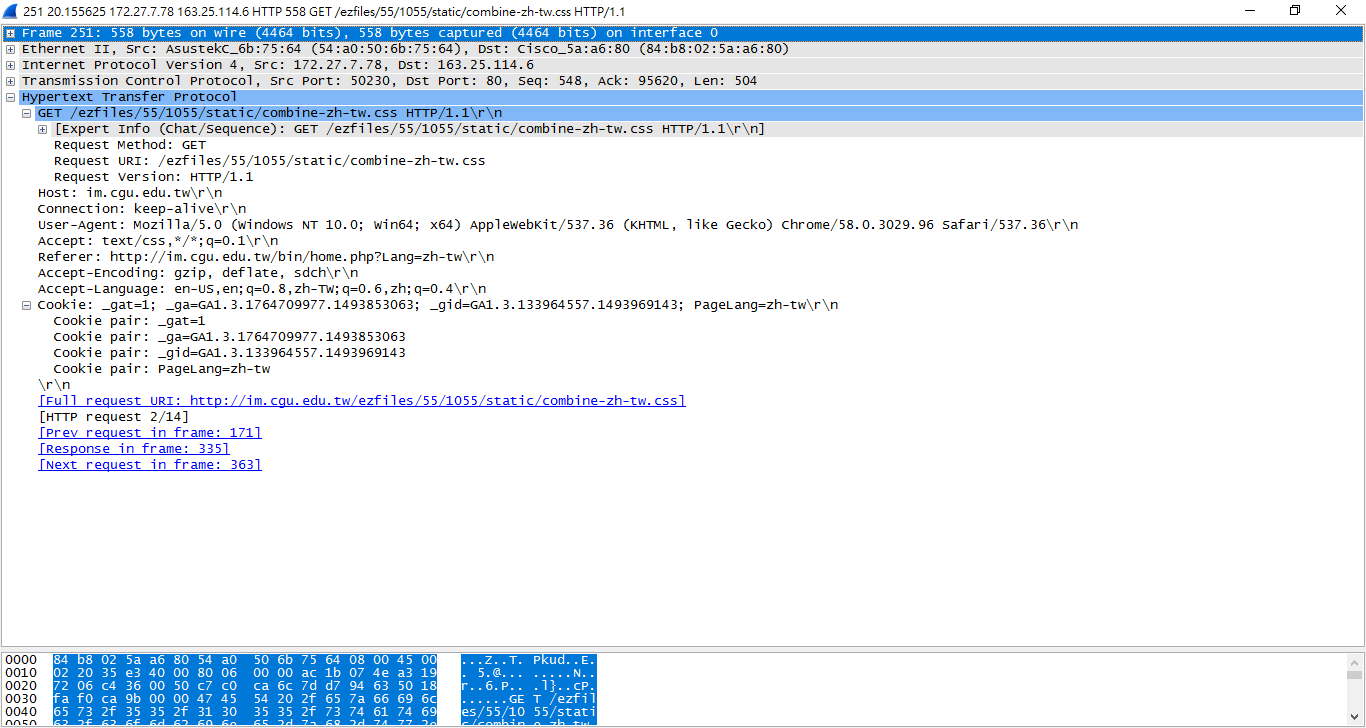
2.Server:則此代表由Apache網頁伺服器所產生。

3.X-Power-By: 表明用於支援當前網頁應用程式的技術（在此為PHP）（版本號細節通常放置在 X-Runtime 或 X-Version 中）。

4. Set-Cookie: HTTP的Cookie

5. Transfer-Encoding: 用來將實體安全地傳輸給用戶的編碼形式。當前定義的方法 包括：分塊(chunked)、壓縮（compress）、縮小（deflate）、壓縮（gzip）、實體（identity）。

6.Content-Type: 當前內容的[MIME](https://zh.wikipedia.org/wiki/MIME)類型(在此是HTML文件)。



->這是NO.251的封包內容，為第二次HTTP請求訊息。

**GET /efziles/55/1055/static/combine-zh-tw.css HTTP/1.1** 為請求行，GET為方法，中間則為URL(請求combine-zh-tw.css這個物件)，後面HTTP/1.1則為HTTP版本。

1.Host:指出該物件所在的主機。

2.Connection: 針對該連線所預期的選項，keep-alive代表瀏覽器是使用永久性連線。

3. User-Agent：瀏覽器類型，如果Servlet返回的內容與瀏覽器類型有關則該值非常有用。

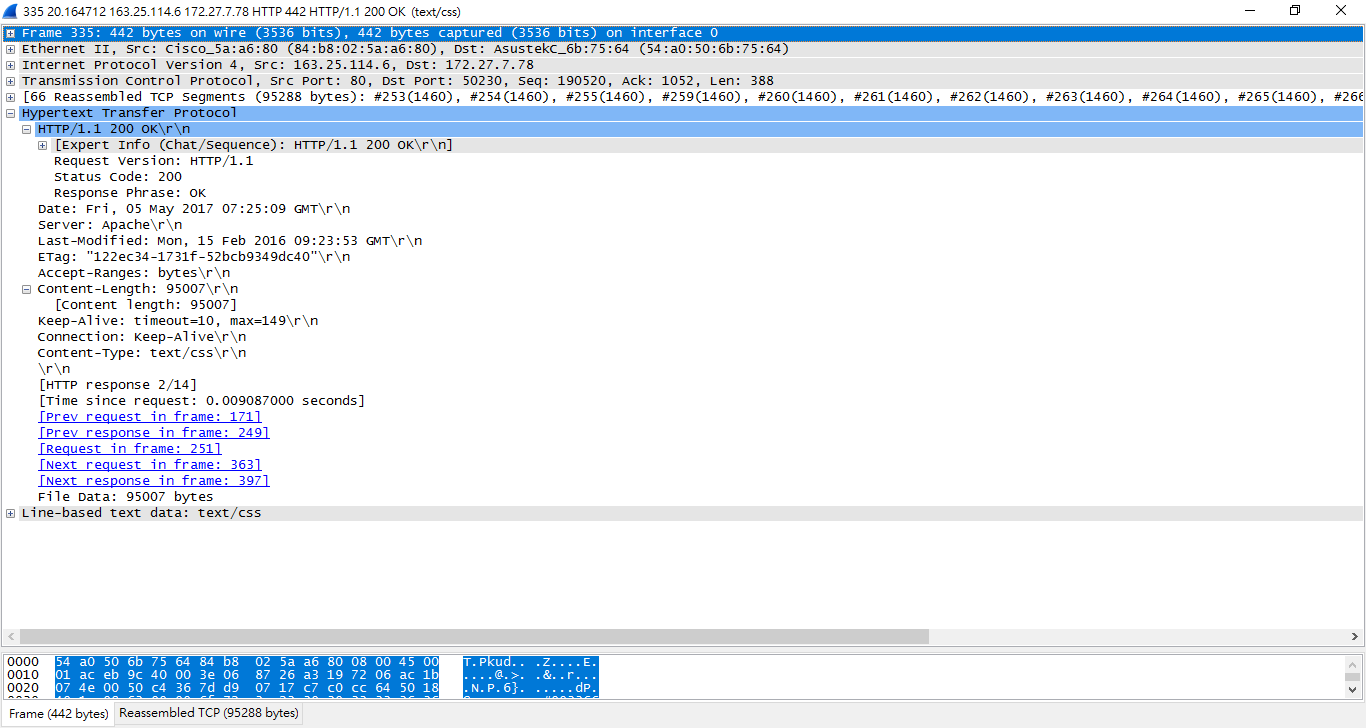
4. Accept:瀏覽器可接受的MIME類型。

5. Referer：包含一個URL，用戶從該URL代表的頁面出發訪問當前請求的頁面。

6. Accept-Encoding：瀏覽器能夠進行解碼的數據編碼方式，比如gzip。Servlet能夠向支持gzip的瀏覽器返回經gzip編碼的HTML頁面。許多情形下這可以減少5到10倍的下載時間。

7. Accept-Language：瀏覽器所希望的語言種類，當服務器能夠提供一種以上的語言版本時要用到。在請求行中有指定為zh-tw(繁體中文)。

8.Cookie: 之前由伺服器通過 Set- Cookie （下文詳述）傳送的一個超文字傳輸協定Cookie。



->這是NO.335的封包內容，為第二次HTTP回應訊息。

**HTTP/1.1 200 OK**  為HTTP狀態行，HTTP/1.1為版本，200為狀態碼，OK為用語。

1.Date: 此條訊息被傳送時的日期和時間(按照 RFC 7231 中定義的「超文字傳輸協定日期」格式來表示)。

2.Server:則此代表由Apache網頁伺服器所產生。

3.Set-Cookie: HTTP的Cookie

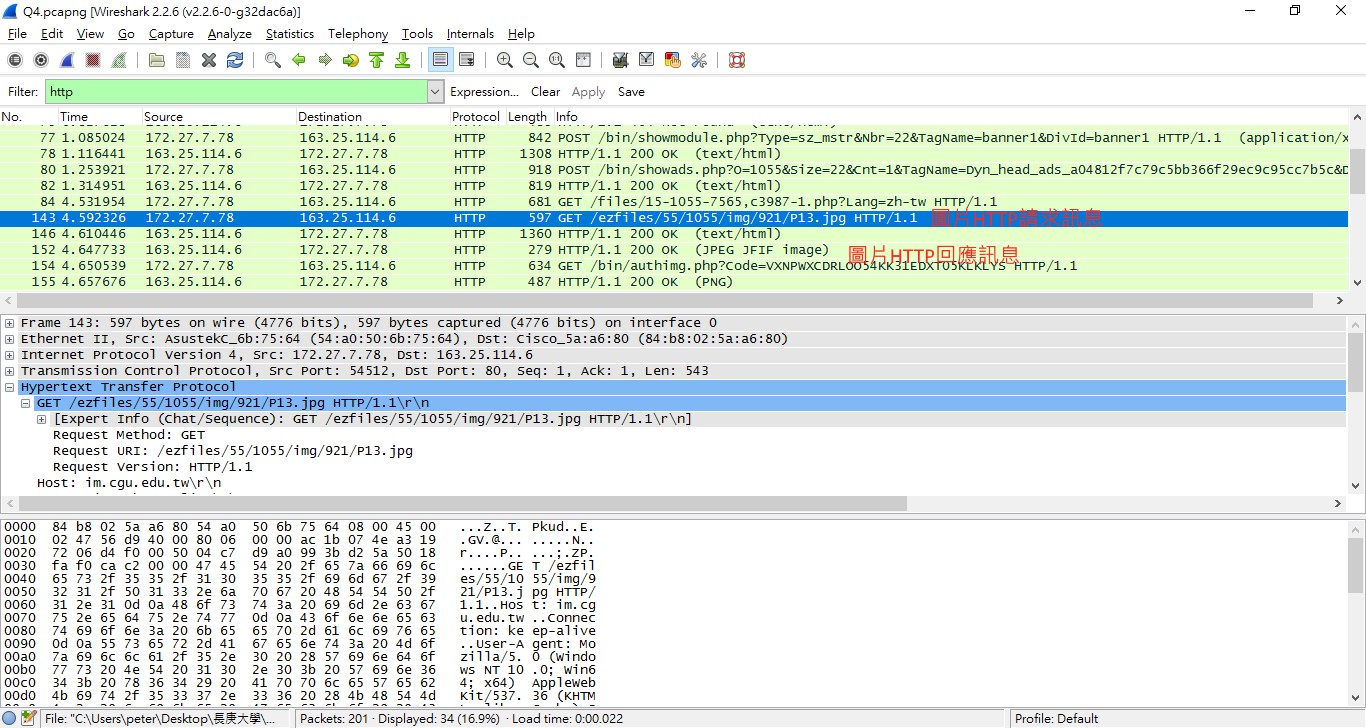
4.Transfer-Encoding: 用來將實體安全地傳輸給用戶的編碼形式。當前定義的方法 包括：分塊(chunked)、壓縮（compress）、縮小（deflate）、壓縮（gzip）、實體（identity）。

5.Content-Type: 當前內容的[MIME](https://zh.wikipedia.org/wiki/MIME)類型(在此是.css樣式表)。

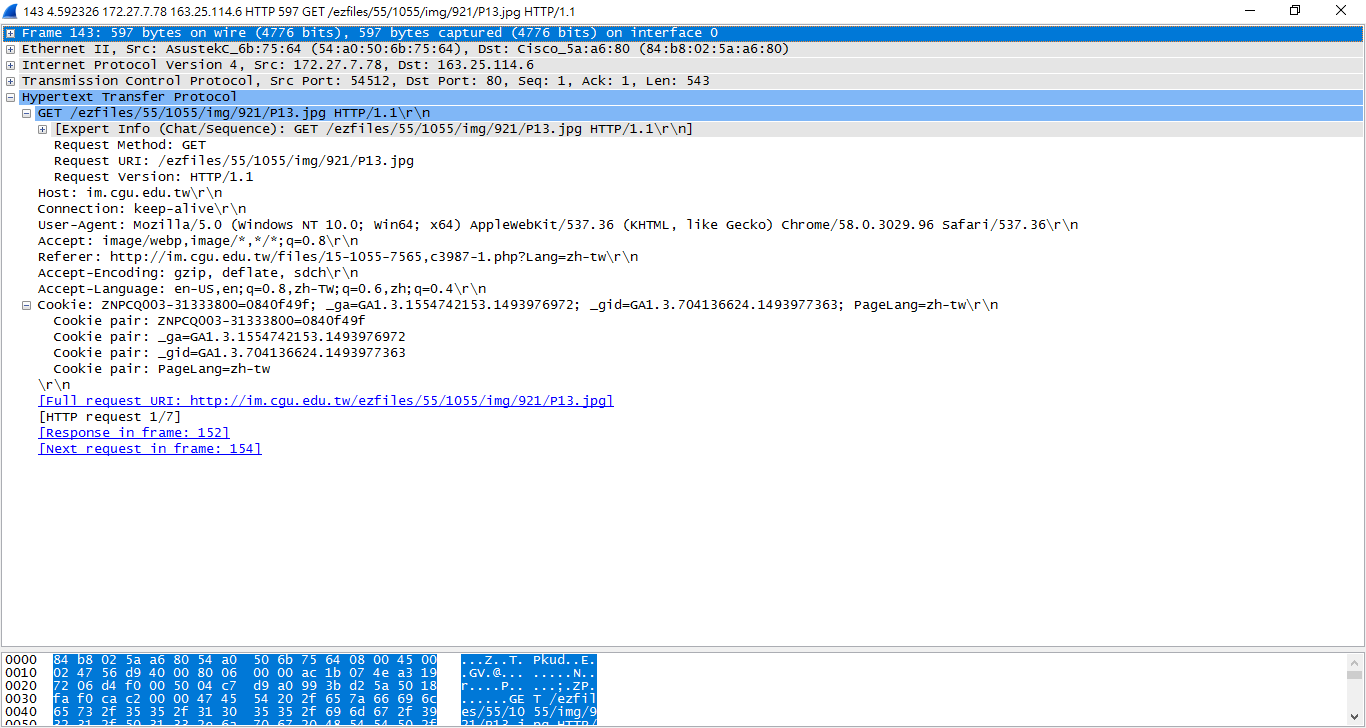
**四、HTML Documents with Embedded Objects：**



->我們選了系上助理教授簡介的網頁，並擷取到了陳昱仁老師大頭照的封包。



->NO.143的封包為圖片的HTTP請求訊息，NO.152的封包為圖片的HTTP回應訊息。



->這是NO.143的封包內容，為圖片的HTTP請求訊息。

**GET /efziles/55/1055/img/921/P13.jpg HTTP/1.1** 為請求行，GET為方法，中間則為URL(請求P13.jpg這個物件)，後面HTTP/1.1則為HTTP版本。

1.Host:指出該物件所在的主機。

2.Connection: 針對該連線所預期的選項，keep-alive代表瀏覽器是使用永久性連線。

3. User-Agent：瀏覽器類型，如果Servlet返回的內容與瀏覽器類型有關則該值非常有用。

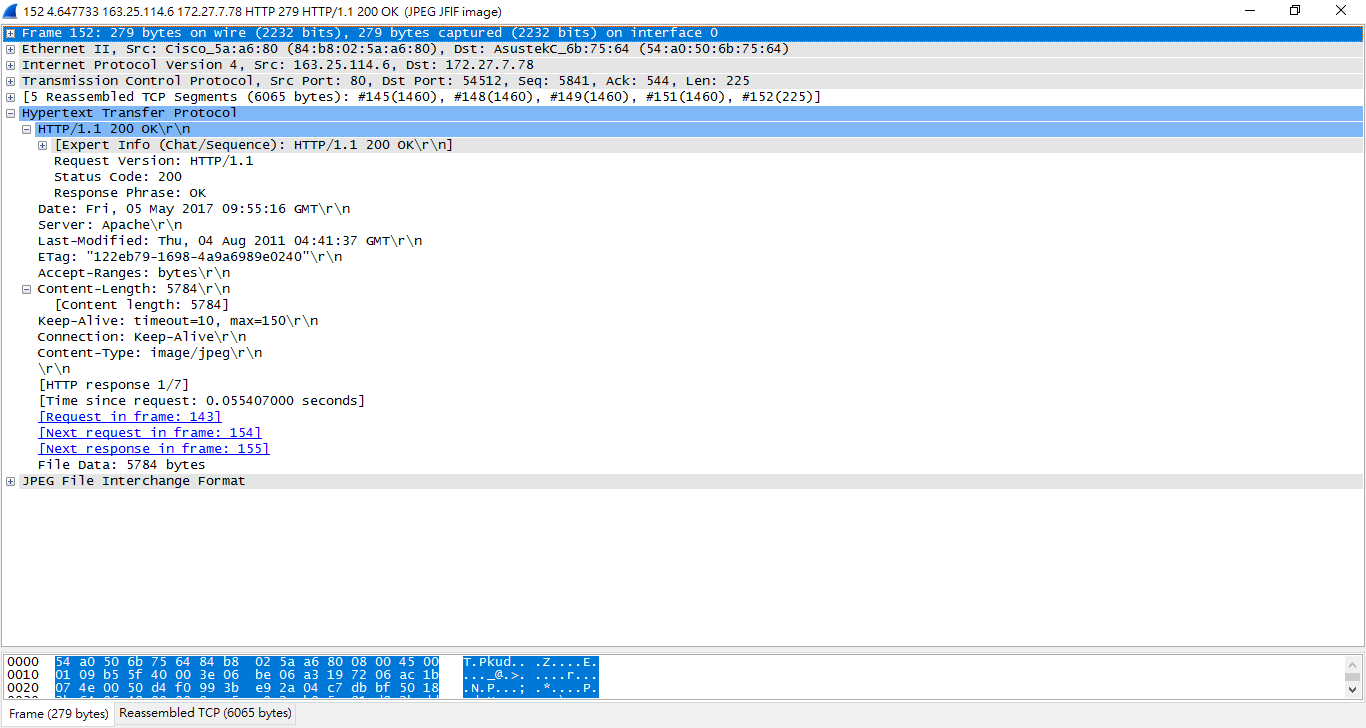
4. Accept:瀏覽器可接受的MIME類型。

5. Referer：包含一個URL，用戶從該URL代表的頁面出發訪問當前請求的頁面。

6. Accept-Encoding：瀏覽器能夠進行解碼的數據編碼方式，比如gzip。Servlet能夠向支持gzip的瀏覽器返回經gzip編碼的HTML頁面。許多情形下這可以減少5到10倍的下載時間。

7. Accept-Language：瀏覽器所希望的語言種類，當服務器能夠提供一種以上的語言版本時要用到。在請求行中有指定為zh-tw(繁體中文)。

8.Cookie: 之前由伺服器通過 Set- Cookie （下文詳述）傳送的一個超文字傳輸協定Cookie。



->這是NO.152的封包內容，為圖片的HTTP回應訊息。

**HTTP/1.1 200 OK**  為HTTP狀態行，HTTP/1.1為版本，200為狀態碼，OK為用語。

1.Date: 此條訊息被傳送時的日期和時間(按照 RFC 7231 中定義的「超文字傳輸協定日期」格式來表示)。

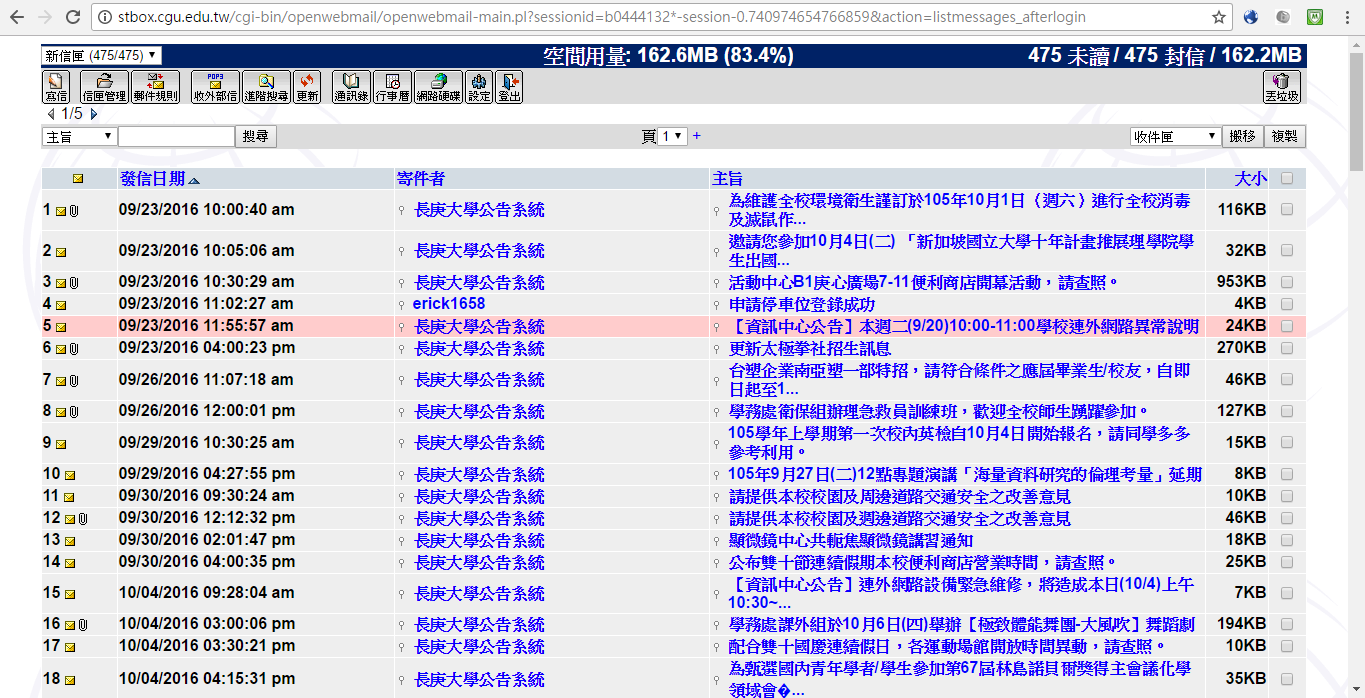
2.Server:則此代表由Apache網頁伺服器所產生。

3.Set-Cookie: HTTP的Cookie

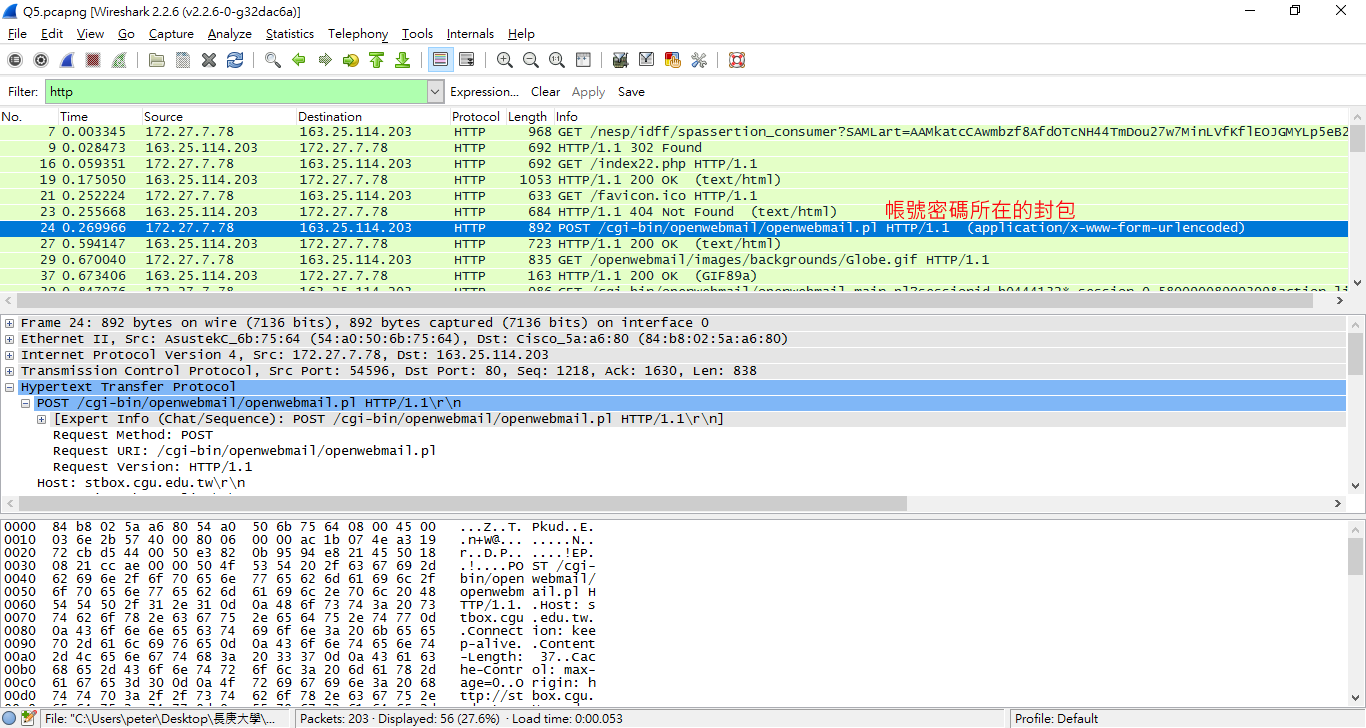
4.Transfer-Encoding: 用來將實體安全地傳輸給用戶的編碼形式。當前定義的方法 包括：分塊(chunked)、壓縮（compress）、縮小（deflate）、壓縮（gzip）、實體（identity）。

5.Content-Type: 當前內容的[MIME](https://zh.wikipedia.org/wiki/MIME)類型(在此是.jpeg)。

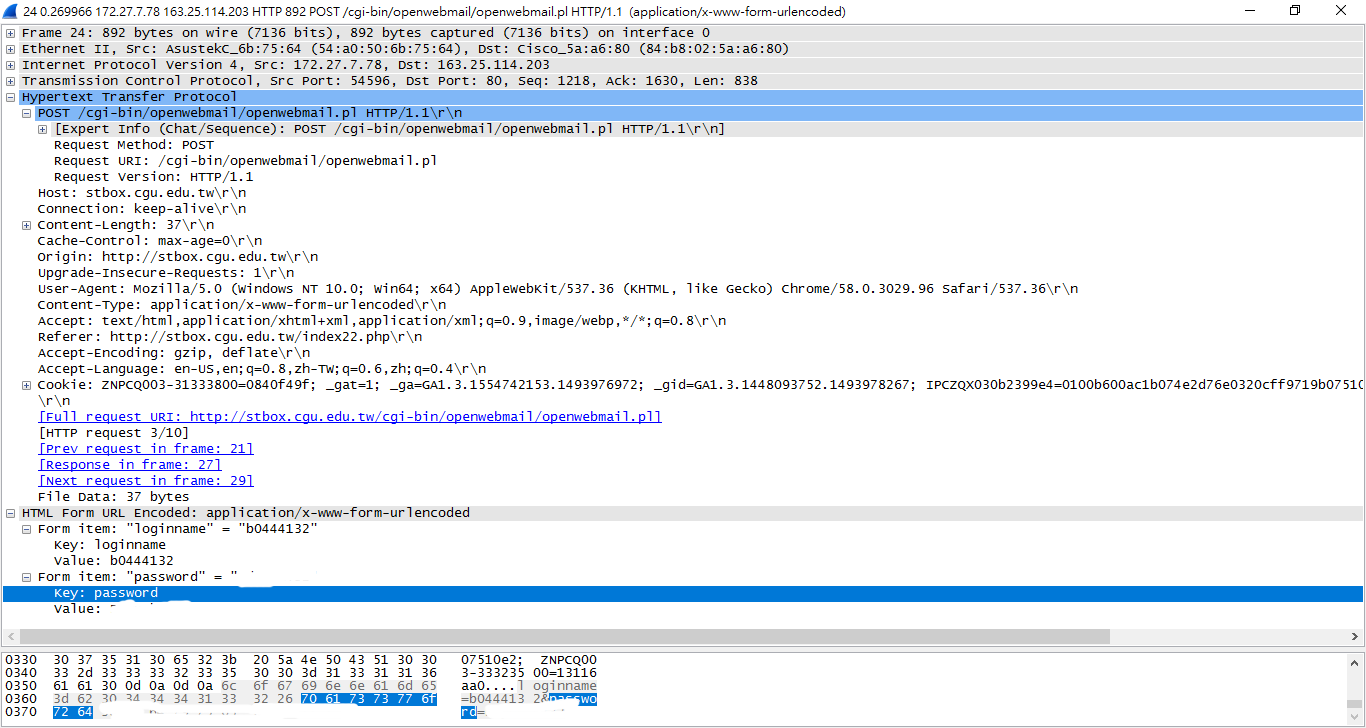
**五、HTTP Authentication：**



->我們選的長庚大學學生Web-mail來進行封包擷取。



->NO.24為記錄帳號密碼的封包。

****

->NO.24封包內容，密碼已被碼掉。

**POST/cgi-bin/operwebmail/openwebmail.pl HTTP/1.1** 為請求行，POST為方法，中間則為URL(請求openwenmail.pl這個物件)，後面HTTP/1.1則為HTTP版本。

1.Host:指出該物件所在的主機。

2.Connection: 針對該連線所預期的選項，keep-alive代表瀏覽器是使用永久性連線。

3. Content-Length: 以八位位元組陣列（8位元的位元組）表示的請求體的長度

4. [Cache-Control](https://zh.wikipedia.org/wiki/%E7%BD%91%E9%A1%B5%E5%BF%AB%E7%85%A7):用來指定在這次的請求/回應鏈中的所有快取機制 都必須 遵守的指令

5. Origin: 發起一個針對 跨來源資源共享 的請求（要求伺服器在回應中加入一個『存取控制-允許來源』（'Access-Control-Allow-Origin'）欄位）。

6. User-Agent：瀏覽器類型，如果Servlet返回的內容與瀏覽器類型有關則該值非常有用。

7. Content-Type:請求體的多媒體類型（用於POST和PUT請求中）

8. Accept:瀏覽器可接受的MIME類型。

9. Referer：包含一個URL，用戶從該URL代表的頁面出發訪問當前請求的頁面。

10. Accept-Encoding：瀏覽器能夠進行解碼的數據編碼方式，比如gzip。Servlet能夠向支持gzip的瀏覽器返回經gzip編碼的HTML頁面。許多情形下這可以減少5到10倍的下載時間。

11. Accept-Language：瀏覽器所希望的語言種類，當服務器能夠提供一種以上的語言版本時要用到。在請求行中有指定為zh-tw(繁體中文)。

12.Cookie: 之前由伺服器通過 Set- Cookie （下文詳述）傳送的一個超文字傳輸協定Cookie。

13.Form Item:顯示帳號密碼的所在地。

五、心得

網路的進步，帶動整個世界的改變，而如今上了電腦網路的課程，讓我們更進一步認識網路架構及運作。HTTP協定可說是現今被廣泛應用的網路協定之一，也是我們一般在瀏覽網頁最容易接觸到的，這次實驗，讓我們不只是單純的「逛」網頁，而是深入其中，分析網頁的封包。透過一個個網站，會發現我們所瀏覽的那些光鮮亮麗的網頁，其實不過是一個個封包所組成，雖然每個封包最多只能儲存約1.5kb大小的資料，但是在網路高速的傳遞下，我們仍可以在很短的時間內，看到一個網頁的完整呈現，封包的切割、傳遞、重組，看起來繁複的多重步驟和那眾多封包的處理，卻能夠在幾毫秒或幾秒的時間內完成，不得不感嘆，網際網路的進步，將許多不可能化為可能，讓我們能更容易取得我們所需的資料，透過網路，滿足商業、娛樂、教育等多種需求