

BÁO CÁO THỰC HÀNH

Môn học: Nhập môn mạng máy tính

Buổi báo cáo: Lab 02

Tên chủ đề: Tên bài thực hành

GVHD: Phan Trung Phát

Ngày thực hiện: 07/10/2022

Ngày nộp báo cáo: 12/10/2022

1. THÔNG TIN CHUNG:

(Liệt kê tất cả các thành viên trong nhóm)

Lớp: IT005.N11.KHTN.1

STT	Họ và tên	MSSV	Email
1	Trương Thanh Minh	21520064	21520064@gm.uit.edu.vn

2. ĐÁNH GIÁ KHÁC:

Nội dung	Kết quả
Tổng thời gian thực hiện bài thực hành trung bình	7 ngày
Link Video thực hiện (nếu có)	
Ý kiến (nếu có) + Khó khăn + Đề xuất ...	
Điểm tự đánh giá	9.5

Phần bên dưới của báo cáo này là báo cáo chi tiết của nhóm thực hiện.

BÁO CÁO CHI TIẾT

1) HTTP GET/response có điều kiện

1. Trình duyệt đang sử dụng phiên bản HTTP 1.0 hay 1.1? Phiên bản HTTP server đang sử dụng là bao nhiêu?

No.	Time	Source	Destination	Protocol	Length	Info
453	1.501728	172.30.160.226	172.30.60.117	HTTP	495	GET /21520142.html HTTP/1.1
466	1.607113	172.30.60.117	172.30.160.226	HTTP	678	HTTP/1.1 200 OK (text/html)
468	1.638386	172.30.160.226	172.30.60.117	HTTP	438	GET /IMG_0880.jpg HTTP/1.1
540	1.708524	172.30.60.117	172.30.160.226	HTTP	960	HTTP/1.1 200 OK (JPEG JFIF image)
611	3.547239	172.30.160.226	172.30.60.117	HTTP	607	GET /21520142.html HTTP/1.1
653	3.769095	172.30.60.117	172.30.160.226	HTTP	197	HTTP/1.1 304 Not Modified

Hình 1: Các gói tin bắt được ở file .html vừa tạo ra sau khi dùng bộ lọc http

```
GET /21520142.html HTTP/1.1
HTTP/1.1 200 OK (text/html)
```

Hình 2: Chi tiết phiên bản HTTP của trình duyệt và server

Ta thấy rằng, sau khi sử dụng bộ lọc http, các gói tin được trả về sẽ chỉ bao gồm các gói tin có protocol là HTTP. Qua hình trên, dễ dàng ta thấy rằng, khi người dùng gửi request lên máy chủ thì trong phần **info** của gói tin đó sẽ bao gồm phiên bản HTTP của trình duyệt mà ta đang sử dụng. Cụ thể, trong ảnh trên, ta biết được rằng, **phiên bản HTTP** của trình duyệt ta đang sử dụng là **HTTP 1.1**.

Cũng dựa vào hình trên, khi ta nhìn vào các gói tin mà server phản hồi cho request mà người dùng gửi lên đều có thông tin là **HTTP/1.1**. Từ đó, ta có thể suy ra là phiên bản HTTP server đang sử dụng là **HTTP 1.1**.

2. Địa chỉ IP của máy tính bạn là bao nhiêu? Của web server là bao nhiêu?

Source	Destination	Protocol	Length	Info
172.30.160.226	172.30.60.117	HTTP	495	GET /21520142.html HTTP/1.1

Hình 3: Chi tiết về địa chỉ IP của máy tính bạn và web server

Khi gửi request từ máy ta đến server thì Source chính là máy ta và Destination chính là máy chủ. Do đó, có thể thấy, **IP của máy ta là 172.30.160.226** và **IP của server là 172.30.60.117**.

3. Mã trạng thái (status code) trả về từ server là gì?

540	1.708524	172.30.60.117	172.30.160.226	HTTP	960	HTTP/1.1 200 OK (JPEG JFIF image)
466	1.607113	172.30.60.117	172.30.160.226	HTTP	678	HTTP/1.1 200 OK (text/html)
653	3.769095	172.30.60.117	172.30.160.226	HTTP	197	HTTP/1.1 304 Not Modified

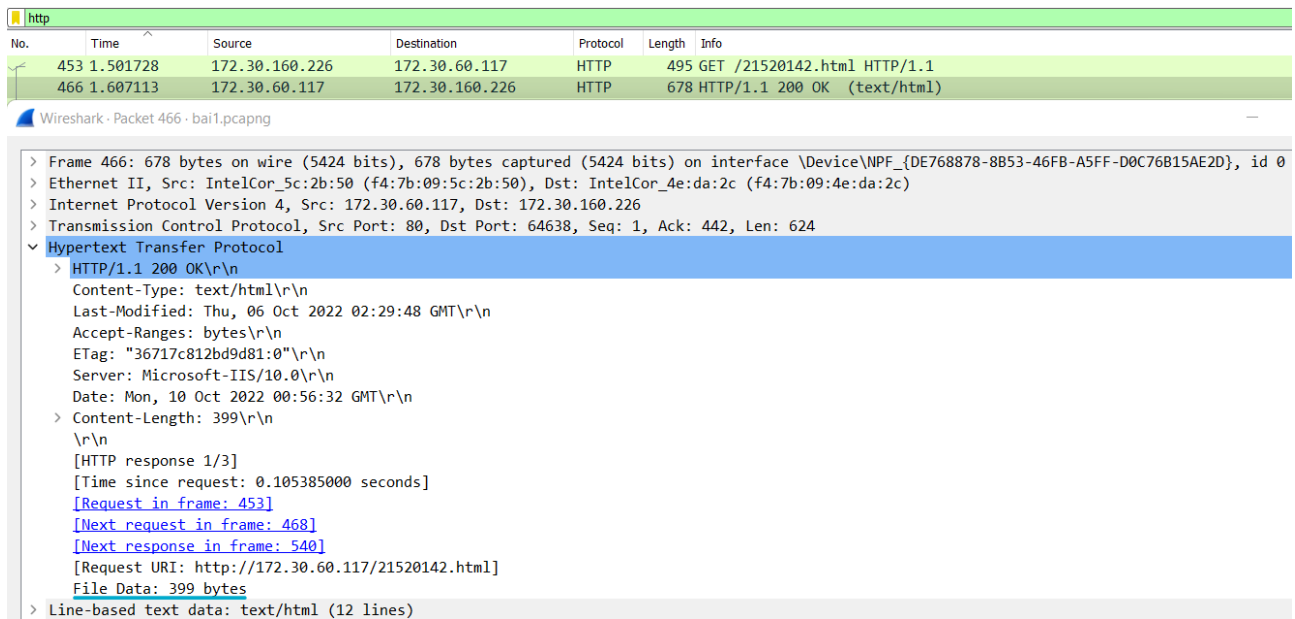
Hình 4: Các gói tin mà server phản hồi

Mã trạng thái trả về từ server là 200 OK.

- 200 OK: Truy cập thành công đến server.

4. Server đã trả về cho trình duyệt bao nhiêu bytes nội dung?

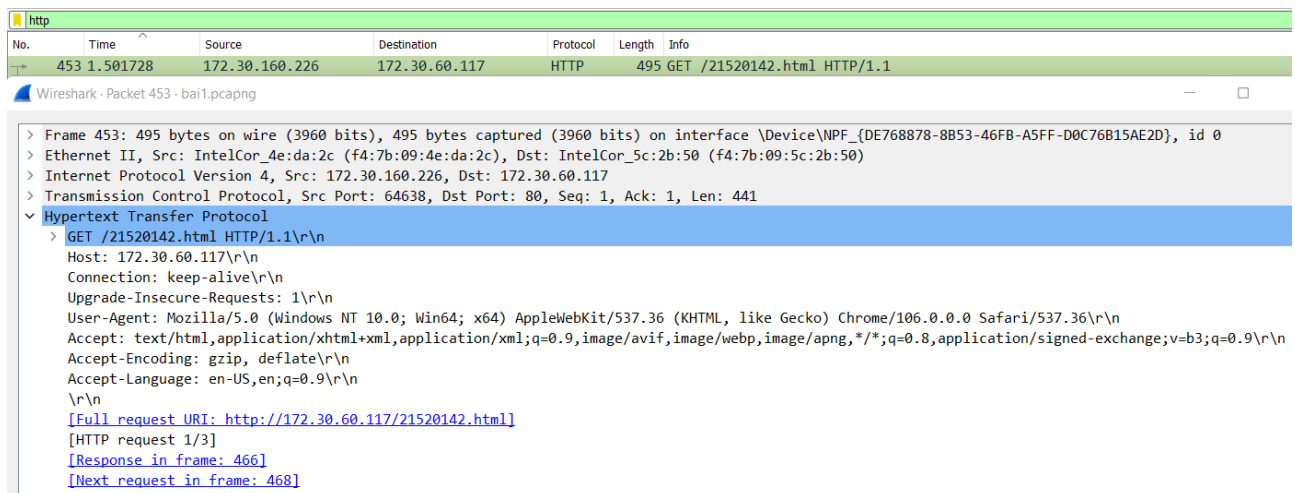
Theo hình 4, server trả về cho ta 1 gói tin: text.



Hình 5: Số bytes ứng với gói tin dạng text đầu tiên

Qua hình ảnh trên, ở **file data** số bytes nội dung của ta là 399 bytes.

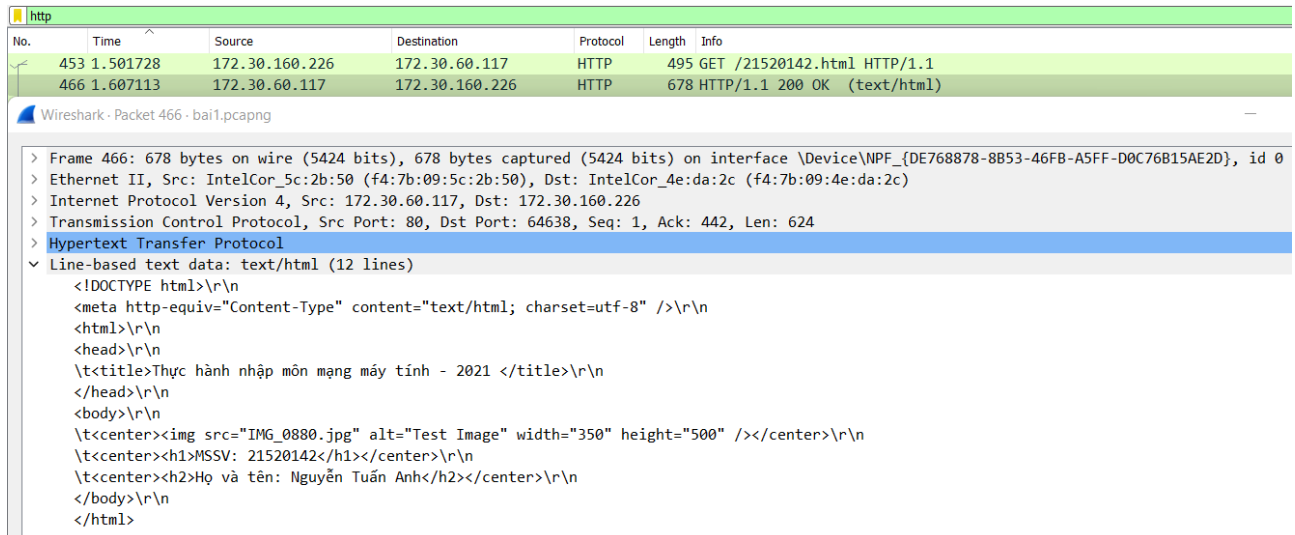
5. Xem xét nội dung của HTTP GET đầu tiên. Bạn có thấy dòng “IF – MODIFIED SINCE” hay không?



Hình 6: Nội dung của HTTP GET đầu tiên

Qua hình ảnh trên, ta thấy rằng, trong nội dung của HTTP GET đầu tiên, không tồn tại dòng “IF – MODIFIED SINCE”.

6. Xem xét nội dung phản hồi từ server. Server có thật sự trả về nội dung của file HTML hay không? Tại sao?



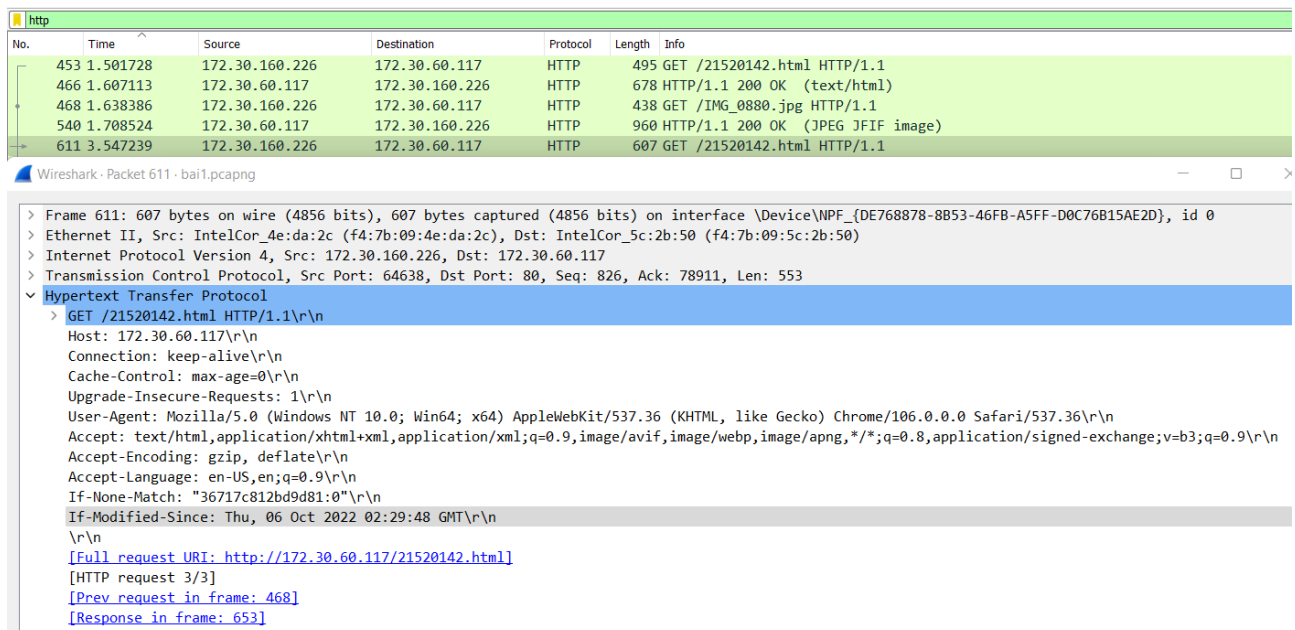
Hình 7: Nội dung phản hồi từ server

Từ hình ảnh trên, sau khi xem nội dung phản hồi của server, ta thấy rằng server đã thật sự trả về nội dung của file HTML.

Giải thích: Vì ban đầu trước khi bắt gói tin, ta đã xóa cache rồi. Do đó, khi người dùng gửi request lên server, server sẽ kiểm tra xem trong cache có nội dung đó chưa. Nếu chưa thì server sẽ trả về nội dung của file đó cho người dùng. Ngược lại thì không. Vì trước khi bắt gói tin, ta đã xóa bộ nhớ cache rồi, nên server sẽ không tìm thấy file đó. Do đó, nội dung của file đó sẽ được trả về cho người dùng.

[Servlets - Server HTTP Response \(tutorialspoint.com\)](http://tutorialspoint.com)

7. Xem xét nội dung của HTTP GET thứ 2. Bạn có thấy dòng “IF-MODIFIED-SINCE” hay không? Nếu có, giá trị của IF-MODIFIED-SINCE là gì?



Hình 8: Nội dung của HTTP GET thứ 2

Trong hình trên, ở dòng được bôi đậm, ta thấy có xuất hiện “IF-MODIFIED-SINCE” với giá trị là: **Thu, 06 Oct 2022 02:29:48 GMT\r\n**.

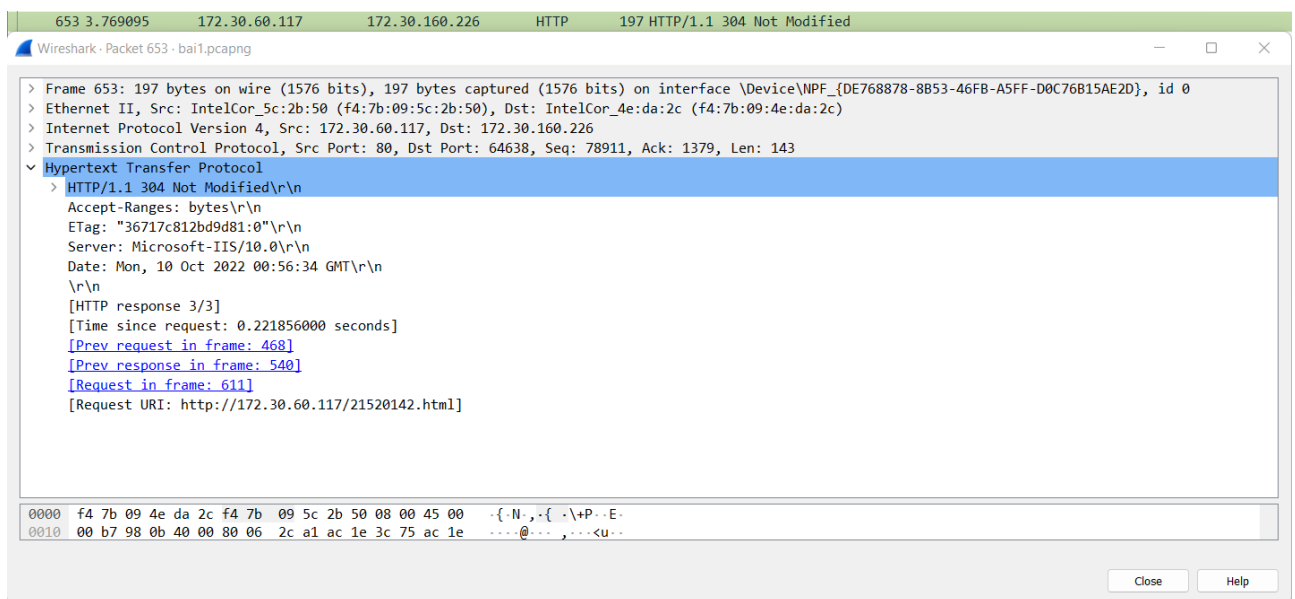
8. Mã trạng thái HTTP được trả về từ server tương ứng với HTTP GET thứ 2 là gì? Ý nghĩa nó là gì? Server có thật sự gửi về nội dung của file hay không? Giải thích.

611	3.547239	172.30.160.226	172.30.60.117	HTTP	607 GET /21520142.html HTTP/1.1
653	3.769095	172.30.60.117	172.30.160.226	HTTP	197 HTTP/1.1 304 Not Modified

Hình 9: Gói tin HTTP GET thứ 2 và gói tin phản hồi của nó

Qua hình trên, ta thấy rằng giá trị trả về từ server tương ứng với HTTP GET thứ 2 là **304 NOT MODIFIED**.

Trạng thái này cho ta biết là nội dung của trang web trên chưa bị sửa đổi hay nói cách khác là nội dung của trang web đó vẫn giống với nội dung của lần request trước đó.



Hình 10: Thông tin của gói tin phản hồi từ server của HTTP GET thứ 2

Trong hình trên, ta thấy phần thông tin của gói tin đó không có chỗ nào hiển thị cho ta về nội dung của trang web như lần GET đầu tiên.

Giải thích: Vì lúc này, trong bộ nhớ cache của ta đã có nội dung của file đó ở lần gửi request đầu tiên (được minh chứng thông qua trạng thái 304 NOT MODIFIED được trả về), do đó, lúc này, server sẽ không gửi lại nội dung đó cho người dùng nữa.

9. Trình duyệt đã gửi bao nhiêu HTTP GET? Đến những địa chỉ IP nào?

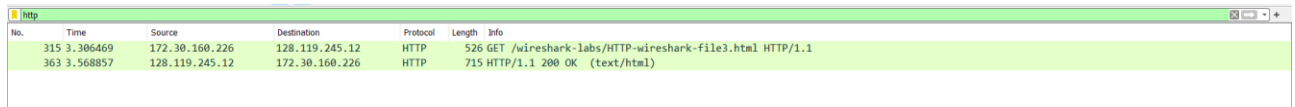
No.	Time	Source	Destination	Protocol	Length	Info
453	1.501728	172.30.160.226	172.30.60.117	HTTP	495	GET /21520142.html HTTP/1.1
611	3.547239	172.30.160.226	172.30.60.117	HTTP	607	GET /21520142.html HTTP/1.1
468	1.638386	172.30.160.226	172.30.60.117	HTTP	438	GET /IMG_0880.jpg HTTP/1.1

Hình 11: Các HTTP GET của trình duyệt

Qua hình trên, ta thấy rằng trình duyệt đã gửi **3** HTTP GET. Và dễ dàng biết được cả 3 request đó đều gửi về **cùng một địa chỉ IP** là: **172.30.60.117**.

2) Truy cập các trang HTTP dài

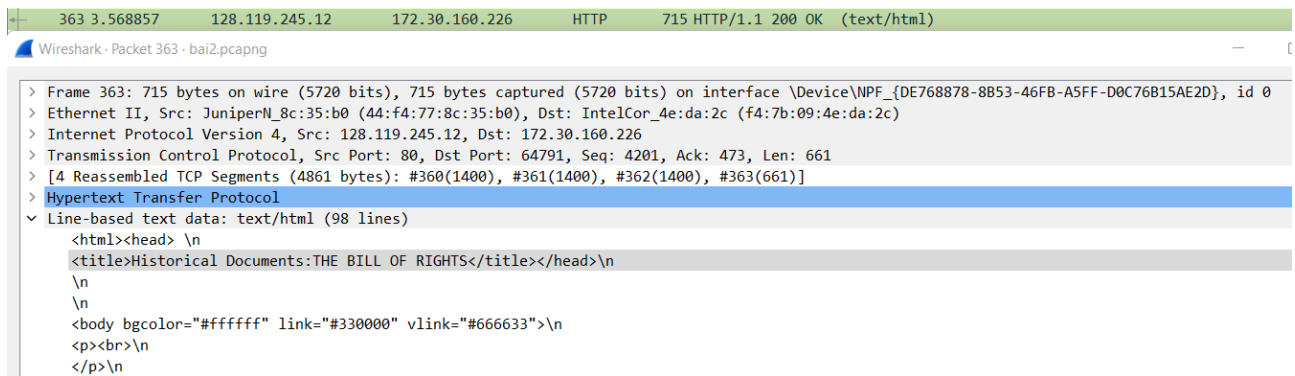
10. Trình duyệt đã gửi bao nhiêu HTTP GET? Dòng “THE BILL OF RIGHTS” được chứa trong gói tin phản hồi thứ mấy?



No.	Time	Source	Destination	Protocol	Length	Info
315	3.306469	172.30.160.226	128.119.245.12	HTTP	526	GET /wireshark-labs/HTTP-wireshark-file3.html HTTP/1.1
363	3.568857	128.119.245.12	172.30.160.226	HTTP	715	HTTP/1.1 200 OK (text/html)

Hình 12: Tất cả các gói tin của web thứ 2 sau khi dùng bộ lọc http

Ta thấy rằng trình duyệt đã gửi 1 HTTP GET và đồng thời cũng chỉ nhận được 1 phản hồi từ server.



No.	Time	Source	Destination	Protocol	Length	Info
363	3.568857	128.119.245.12	172.30.160.226	HTTP	715	HTTP/1.1 200 OK (text/html)

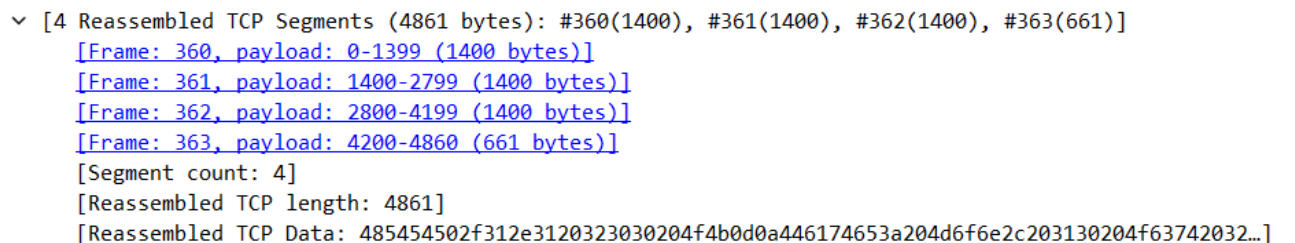
```

> Frame 363: 715 bytes on wire (5720 bits), 715 bytes captured (5720 bits) on interface \Device\NPF_{DE768878-8B53-46FB-A5FF-D0C76B15AE2D}, id 0
> Ethernet II, Src: Juniper_N_8c:35:b0 (44:f4:77:8c:35:b0), Dst: IntelCor_4e:da:2c (f4:7b:09:4e:da:2c)
> Internet Protocol Version 4, Src: 128.119.245.12, Dst: 172.30.160.226
> Transmission Control Protocol, Src Port: 80, Dst Port: 64791, Seq: 4201, Ack: 473, Len: 661
> [4 Reassembled TCP Segments (4861 bytes): #360(1400), #361(1400), #362(1400), #363(661)]
> Hypertext Transfer Protocol
> Line-based text data: text/html (98 lines)
  <html><head> \n
    <title>Historical Documents:THE BILL OF RIGHTS</title></head>\n
    \n
    \n
    <body bgcolor="#ffffff" link="#330000" vlink="#666633">\n
      <p><br>\n
    </p>\n
  
```

Hình 13: Nội dung của gói tin phản hồi đầu tiên của server

Ở dòng được bôi đen ở hình trên, ta thấy rằng dòng “THE BILL OF RIGHTS” xuất hiện ngay ở gói tin phản hồi đầu tiên.

11. Cần bao nhiêu TCP segments để chứa hết HTTP response và nội dung của The Bill of Rights?



```

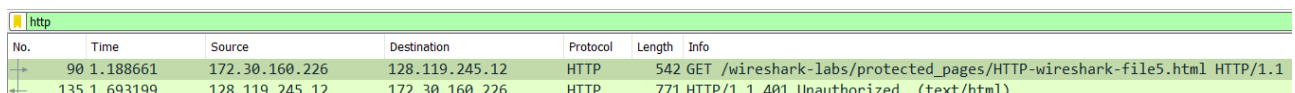
[4 Reassembled TCP Segments (4861 bytes): #360(1400), #361(1400), #362(1400), #363(661)]
[Frame: 360, payload: 0-1399 (1400 bytes)]
[Frame: 361, payload: 1400-2799 (1400 bytes)]
[Frame: 362, payload: 2800-4199 (1400 bytes)]
[Frame: 363, payload: 4200-4860 (661 bytes)]
[Segment count: 4]
[Reassembled TCP length: 4861]
[Reassembled TCP Data: 485454502f312e3120323030204f4b0d0a446174653a204d6f6e2c203130204f63742032...]
  
```

Hình 14: Số TCP segments cần để chứa HTTP response và nội dung của The Bill of Rights

Ta có thể thấy Segment count ở trên là 4. Do đó, ta cần 4 TCP segments để chứa hết HTTP response và nội dung của The Bill of Rights.

3) Chứng thực HTTP

12. Mã trạng thái và ý nghĩa nó trong HTTP response tương ứng với HTTP GET đầu tiên là gì?



No.	Time	Source	Destination	Protocol	Length	Info
90	1.188661	172.30.160.226	128.119.245.12	HTTP	542	GET /wireshark-labs/protected_pages/HTTP-wireshark-file5.html HTTP/1.1
135	1.693199	128.119.245.12	172.30.160.226	HTTP	771	HTTP/1.1 401 Unauthorized (text/html)

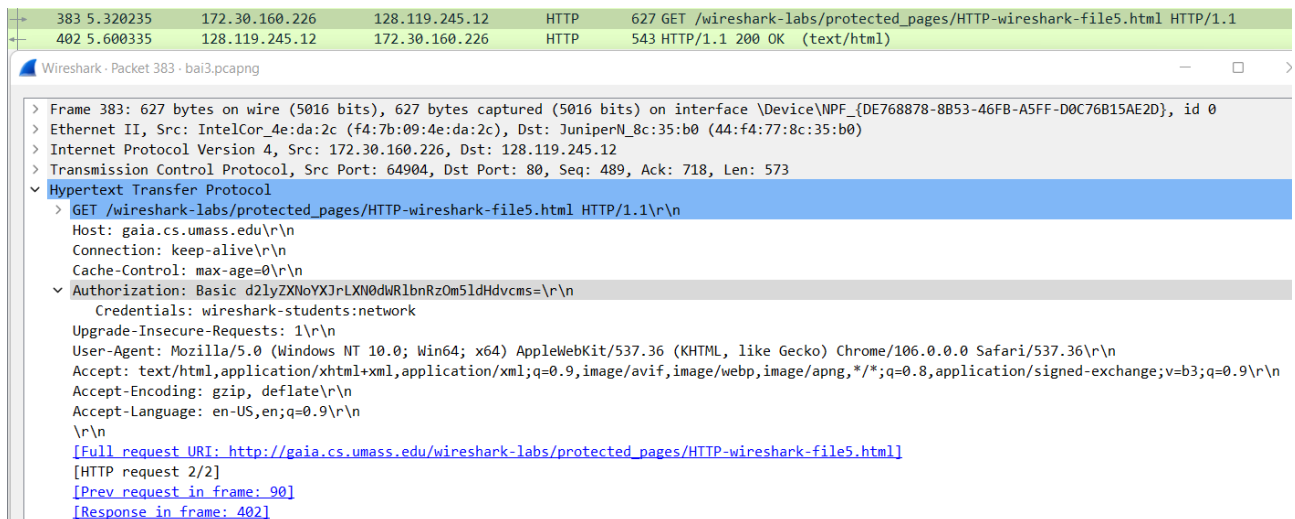
Hình 15: HTTP GET đầu tiên và response tương ứng



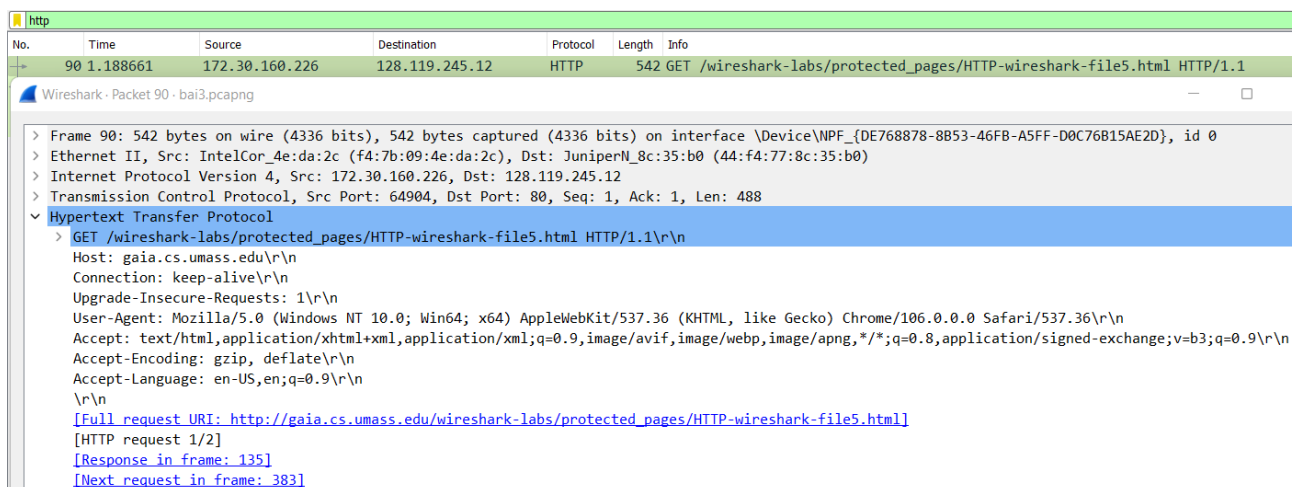
Từ hình trên, ta thấy HTTP response tương ứng với HTTP GET đầu tiên là **401 Unauthorized**.

Mã trạng thái 401 Unauthorized cho ta biết trang web đó yêu cầu thông tin đăng nhập của người dùng. Do đó, response trên trả về 401 Unauthorized vì ban đầu ta chưa nhập username và password tương ứng.

13. Khi trình duyệt gửi HTTP GET lần thứ 2, trường dữ liệu nào mới nào xuất hiện trong HTTP GET?



Hình 16: Nội dung của gói tin HTTP GET lần thứ 2



Hình 17: Nội dung của gói tin HTTP GET lần thứ 1

Khi so sánh giữa nội dung gói tin HTTP GET lần thứ 1 và lần thứ 2, ta thấy rằng, trong nội dung của HTTP GET lần thứ 2 xuất hiện trường dữ liệu mới: **Authorization**.

Trong trường dữ liệu mới đó, ta thấy nội dung **Credentials** nó lưu giữ thông tin **username** và **password** mà ta phải nhập vào nếu muốn truy cập vào trang web.