[1 차 과제 보고서]

HTTP 분석

미디어학과 201821048 이서영

1. 관찰 환경 및 방법

장소 : 집에서 와이파이를 이용해 인터넷에 연결했다.

시간 : 사이트에서 F5 를 눌러 약 10 초 이내의 패킷을 캡쳐했다.

사용한 기능 : 와이어샤크에서 http 패킷만을 분석하기 위해 필터링 기능을 사용했다.

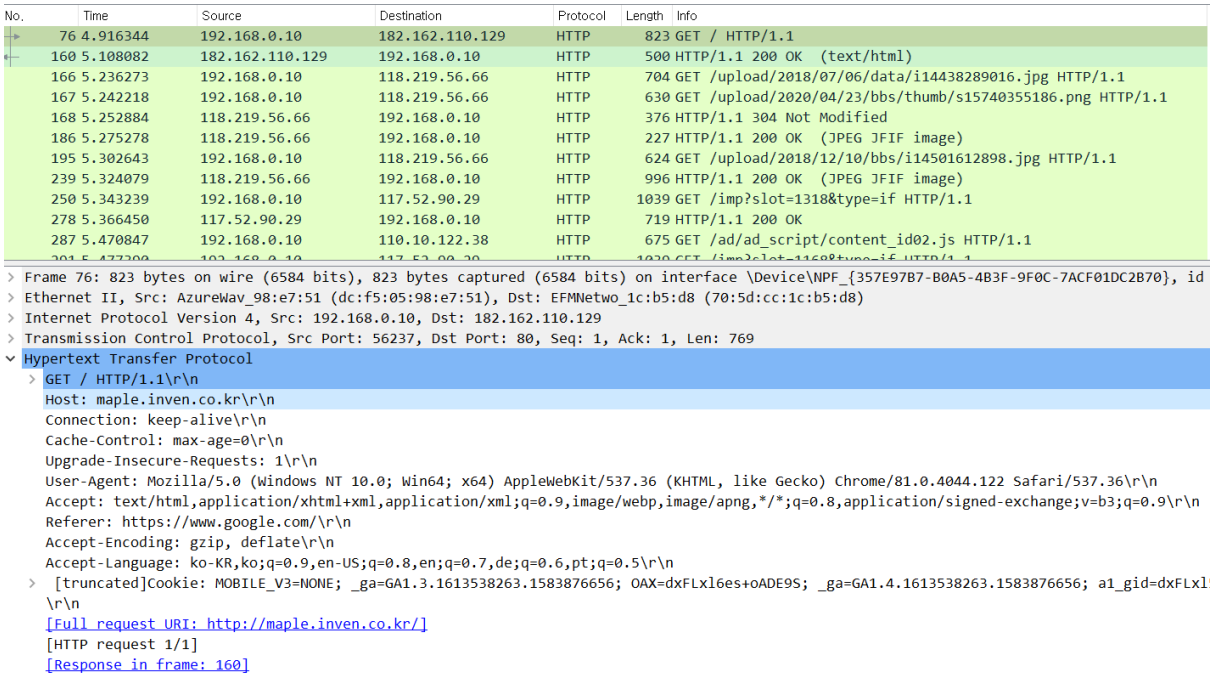
실습 방법 :
1. 새로운 창을 열었다.
2. http 로 된 사이트를 찾았다.
3. 와이어샤크의 캡쳐 기능을 사용했다.
4. 웹사이트를 새로고침 해 원하는 패킷(HTTP)을 기록했다.
5. 와이어샤크 캡쳐기능을 멈추고 저장했다.
5. http 만을 분석하기 위해 필터에 http 를 입력해 해당 내용만을 정렬했다.
6. 각 패킷의 세부내용을 펼쳐 관찰했다.

2. Protocol 분석

1) HTTP request 와 response message 의 header 분석

Request Header : 요청하는 페이지의 주소와 현제 컴퓨터의 정보가 전송된다.
Request Body : POST 요청 시 전송되는 데이터가 들어간다. GET 요청 때는 빈킨이다.
Response Header : 응답 페이지의 상태와 서버에 관한 정보가 전송된다.
Response Body : 페이지의 HTML 소스가 전송된다.

첫번째로 교환한 HTTP Request Header 의 자세한 내용이다.

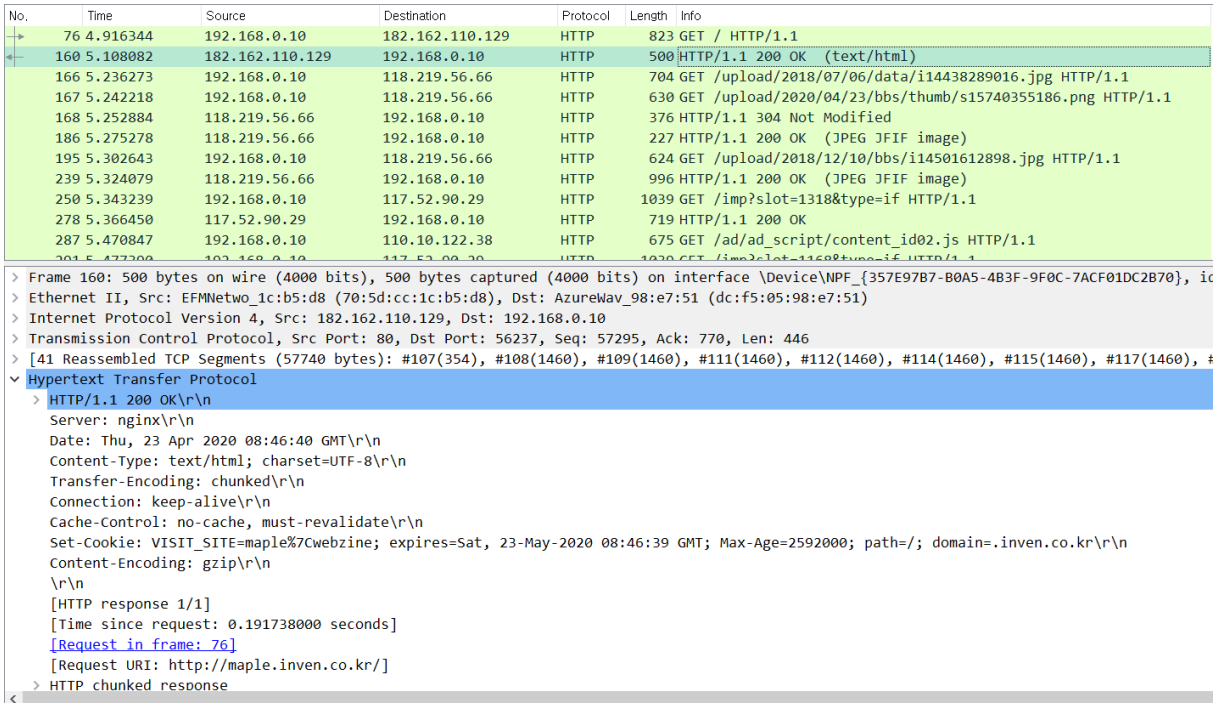


No.	Time	Source	Destination	Protocol	Length	Info
76	4.916344	192.168.0.10	182.162.110.129	HTTP	823	GET / HTTP/1.1
160	5.108082	182.162.110.129	192.168.0.10	HTTP	500	HTTP/1.1 200 OK (text/html)
166	5.236273	192.168.0.10	118.219.56.66	HTTP	704	GET /upload/2018/07/06/data/i14438289016.jpg HTTP/1.1
167	5.242218	192.168.0.10	118.219.56.66	HTTP	630	GET /upload/2020/04/23/bbs/thumb/s15740355186.png HTTP/1.1
168	5.252884	118.219.56.66	192.168.0.10	HTTP	376	HTTP/1.1 304 Not Modified
186	5.275278	118.219.56.66	192.168.0.10	HTTP	227	HTTP/1.1 200 OK (JPEG JFIF image)
195	5.302643	192.168.0.10	118.219.56.66	HTTP	624	GET /upload/2018/12/10/bbs/i14501612898.jpg HTTP/1.1
239	5.324079	118.219.56.66	192.168.0.10	HTTP	996	HTTP/1.1 200 OK (JPEG JFIF image)
250	5.343239	192.168.0.10	117.52.90.29	HTTP	1039	GET /imp?slot=1318&type=if HTTP/1.1
278	5.366450	117.52.90.29	192.168.0.10	HTTP	719	HTTP/1.1 200 OK
287	5.470847	192.168.0.10	110.10.122.38	HTTP	675	GET /ad/ad_script/content_id02.js HTTP/1.1

```
> Frame 76: 823 bytes on wire (6584 bits), 823 bytes captured (6584 bits) on interface \Device\NPF_{357E97B7-B0A5-4B3F-9F0C-7ACF01DC2B70}, id
> Ethernet II, Src: AzureWav_98:e7:51 (dc:f5:05:98:e7:51), Dst: EFMNetwo_1c:b5:d8 (70:5d:cc:1c:b5:d8)
> Internet Protocol Version 4, Src: 192.168.0.10, Dst: 182.162.110.129
> Transmission Control Protocol, Src Port: 56237, Dst Port: 80, Seq: 1, Ack: 1, Len: 769
v Hypertext Transfer Protocol
  > GET / HTTP/1.1\r\n
    Host: maple.inven.co.kr\r\n
    Connection: keep-alive\r\n
    Cache-Control: max-age=0\r\n
    Upgrade-Insecure-Requests: 1\r\n
    User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/81.0.4044.122 Safari/537.36\r\n
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9\r\n
    Referer: https://www.google.com/\r\n
    Accept-Encoding: gzip, deflate\r\n
    Accept-Language: ko-KR,ko;q=0.9,en-US;q=0.8,en;q=0.7,de;q=0.6,pt;q=0.5\r\n
  > [truncated]Cookie: MOBILE_V3=NONE; _ga=GA1.3.1613538263.1583876656; OAX=dxFLxl6es+oADE9S; _ga=GA1.4.1613538263.1583876656; a1_gid=dxFLxl
    \r\n
    [Full request URI: http://maple.inven.co.kr/]
    [HTTP request 1/1]
    [Response in frame: 160]
```

1. GET : 명령어로 페이지를 가져오려고 한다. http 1.1 버전이다.
2. HOST : 도메인은 maple.inven.co.kr 이다.
3. Connection : Keep-alive 로 서버에서 정한 시간만큼 연결을 유지한다.
4. Cache-Control : 서버측에 대한 캐시 컨트롤 요청이다.
5. User-Agent : 브라우저, 운영체제에 대한 정보가 들어있다.
6. Accept : 현재 브라우저가 요청하는 파일의 mime type 이 기록된다.
7. Referer : HTTP 요청을 시도한 페이지의 URL 이다. 어떤 페이지를 통해 현재 페이지로 들어왔는지 정보를 제공한다.
8. Accept-Encoding : 브라우저가 받아들일 수 있는 압축 알고리즘이 전송된다.
9. Accept-Language : 사용자가 이해할 수 있는 자연어의 종류가 전송된다.
10. Cookie : 쿠키 값이 전송된다.
11. 요청 URL 은 서버에서 파일의 위치이다.

Response Header 의 자세한 내용이다.



No.	Time	Source	Destination	Protocol	Length	Info
76	4.916344	192.168.0.10	182.162.110.129	HTTP	823	GET / HTTP/1.1
160	5.108082	182.162.110.129	192.168.0.10	HTTP	500	HTTP/1.1 200 OK (text/html)
166	5.236273	192.168.0.10	118.219.56.66	HTTP	704	GET /upload/2018/07/06/data/i14438289016.jpg HTTP/1.1
167	5.242218	192.168.0.10	118.219.56.66	HTTP	630	GET /upload/2020/04/23/bbs/thumb/s15740355186.png HTTP/1.1
168	5.252884	118.219.56.66	192.168.0.10	HTTP	376	HTTP/1.1 304 Not Modified
186	5.275278	118.219.56.66	192.168.0.10	HTTP	227	HTTP/1.1 200 OK (JPEG JFIF image)
195	5.302643	192.168.0.10	118.219.56.66	HTTP	624	GET /upload/2018/12/10/bbs/i14501612898.jpg HTTP/1.1
239	5.324079	118.219.56.66	192.168.0.10	HTTP	996	HTTP/1.1 200 OK (JPEG JFIF image)
250	5.343239	192.168.0.10	117.52.90.29	HTTP	1039	GET /imp?slot=1318&type=if HTTP/1.1
278	5.366450	117.52.90.29	192.168.0.10	HTTP	719	HTTP/1.1 200 OK
287	5.470847	192.168.0.10	110.10.122.38	HTTP	675	GET /ad/ad_script/content_id02.js HTTP/1.1

```
> Frame 160: 500 bytes on wire (4000 bits), 500 bytes captured (4000 bits) on interface \Device\NPF_{357E97B7-B0A5-4B3F-9F0C-7ACF01DC2B70}, id
> Ethernet II, Src: EFMNetwo_1c:b5:d8 (70:5d:cc:1c:b5:d8), Dst: AzureWav_98:e7:51 (dc:f5:05:98:e7:51)
> Internet Protocol Version 4, Src: 182.162.110.129, Dst: 192.168.0.10
> Transmission Control Protocol, Src Port: 80, Dst Port: 56237, Seq: 57295, Ack: 770, Len: 446
> [41 Reassembled TCP Segments (57740 bytes): #107(354), #108(1460), #109(1460), #111(1460), #112(1460), #114(1460), #115(1460), #117(1460), #
v Hypertext Transfer Protocol
  > HTTP/1.1 200 OK\r\n
    Server: nginx\r\n
    Date: Thu, 23 Apr 2020 08:46:40 GMT\r\n
    Content-Type: text/html; charset=UTF-8\r\n
    Transfer-Encoding: chunked\r\n
    Connection: keep-alive\r\n
    Cache-Control: no-cache, must-revalidate\r\n
    Set-Cookie: VISIT_SITE=maple%7Cwebzine; expires=Sat, 23-May-2020 08:46:39 GMT; Max-Age=2592000; path=/; domain=.inven.co.kr\r\n
    Content-Encoding: gzip\r\n
    \r\n
    [HTTP response 1/1]
    [Time since request: 0.191738000 seconds]
    [Request in frame: 76]
    [Request URI: http://maple.inven.co.kr/]
  > HTTP chunked response
```

1. 200 OK : 정상적으로 응답이 처리되었다.
2. Server : 응답을 담당하는 웹 서버의 종류를 나타낸다.
3. Data : 웹 서버의 현재 시간을 나타낸다.
4. Content-Type : 응답의 mime-type 을 나타낸다.
5. Connection : 소켓 연결을 유지할지 결정한다.
6. Cache-Control : 브라우저 쪽의 캐싱 여부를 결정한다.
7. Set-Cookie : 사용자의 브라우저에 쿠키값을 설정한다.

2) Persistent connection

Pipelining : 여러 세션을 연결한 후 여러 개의 request 를 받아온다.

166	5.236273	192.168.0.10	118.219.56.66	HTTP	704 GET /upload/2018/07/06/data/i14438289016.jpg HTTP/1.1
167	5.242218	192.168.0.10	118.219.56.66	HTTP	630 GET /upload/2020/04/23/bbs/thumb/s15740355186.png HTTP/1.1
168	5.252884	118.219.56.66	192.168.0.10	HTTP	376 HTTP/1.1 304 Not Modified
186	5.275278	118.219.56.66	192.168.0.10	HTTP	227 HTTP/1.1 200 OK (JPEG JFIF image)
195	5.302643	192.168.0.10	118.219.56.66	HTTP	624 GET /upload/2018/12/10/bbs/i14501612898.jpg HTTP/1.1
239	5.324079	118.219.56.66	192.168.0.10	HTTP	996 HTTP/1.1 200 OK (JPEG JFIF image)

HTTP Connetcion 은 Short-lived connection, Persistent connection, Pipelining 3 가지 종류가 있다.

(: establish connection

) : close connection

->, -< : 패킷 전송

Short-lived connection : (->-<)(->-<)(->-<)

Persistent connection : ->-<-->-<-->-<-

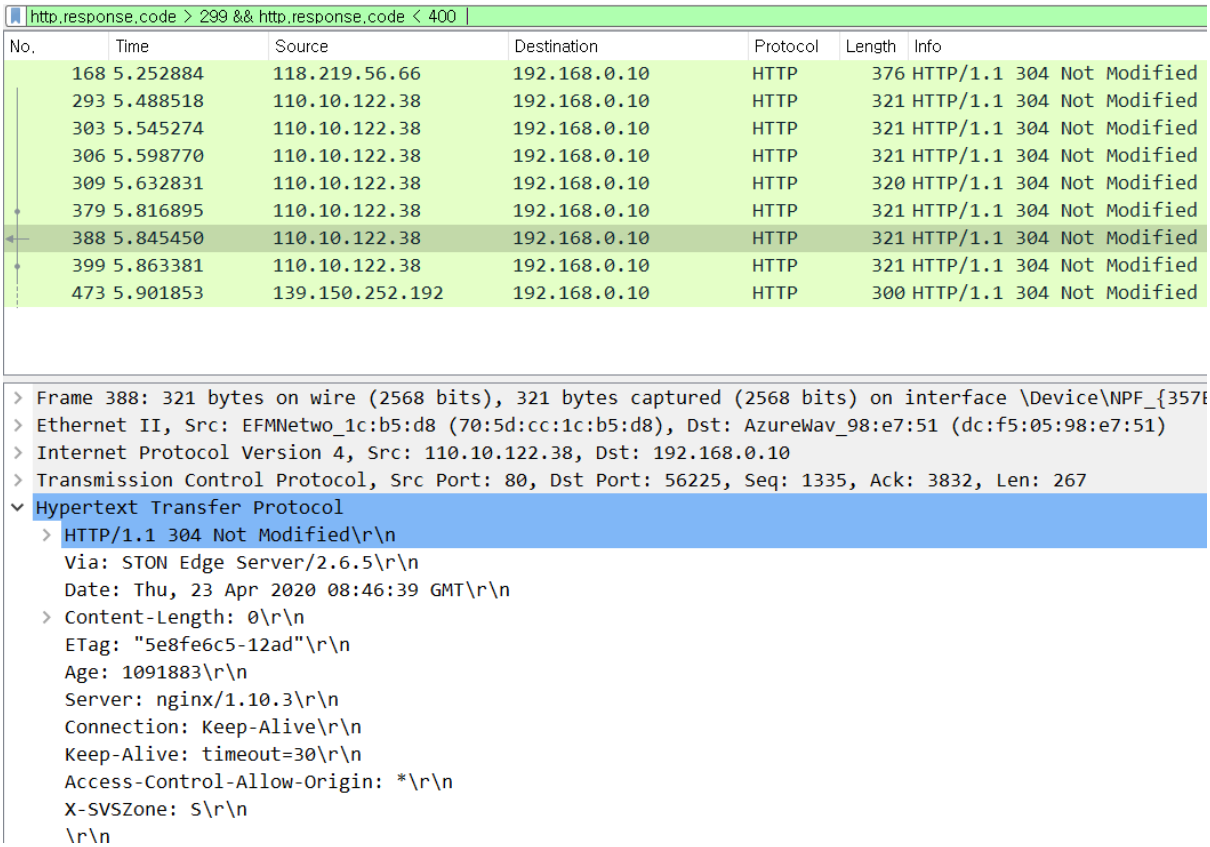
Pipelining : ->->->><-<-<-

직관적으로 보이듯이 파이프라인 방식을 사용하면 같은 양의 정보를 전달할 때 훨씬 빠르게 전달할 수 있다.

3) Redirection 사례

필터에 http.response.code > 299 && http.response.code < 400 을 적용해 redirction 과 관련된 내용을 검색해 보았다. 하지만 304 만 나와 관찰할 수 없었다.

304 Not Modified : URL 이 변경되지 않는 모습을 보였다.



3xx 은 Redirection 코드이다. 각 코드가 나타내는 내용은 다음과 같다.

300 Multiple Choices [RFC7231, Section 6.4.1]

301 Moved Permanently [RFC7231, Section 6.4.2]

302 Found [RFC7231, Section 6.4.3]

303 See Other [RFC7231, Section 6.4.4]

304 Not Modified [RFC7232, Section 4.1]

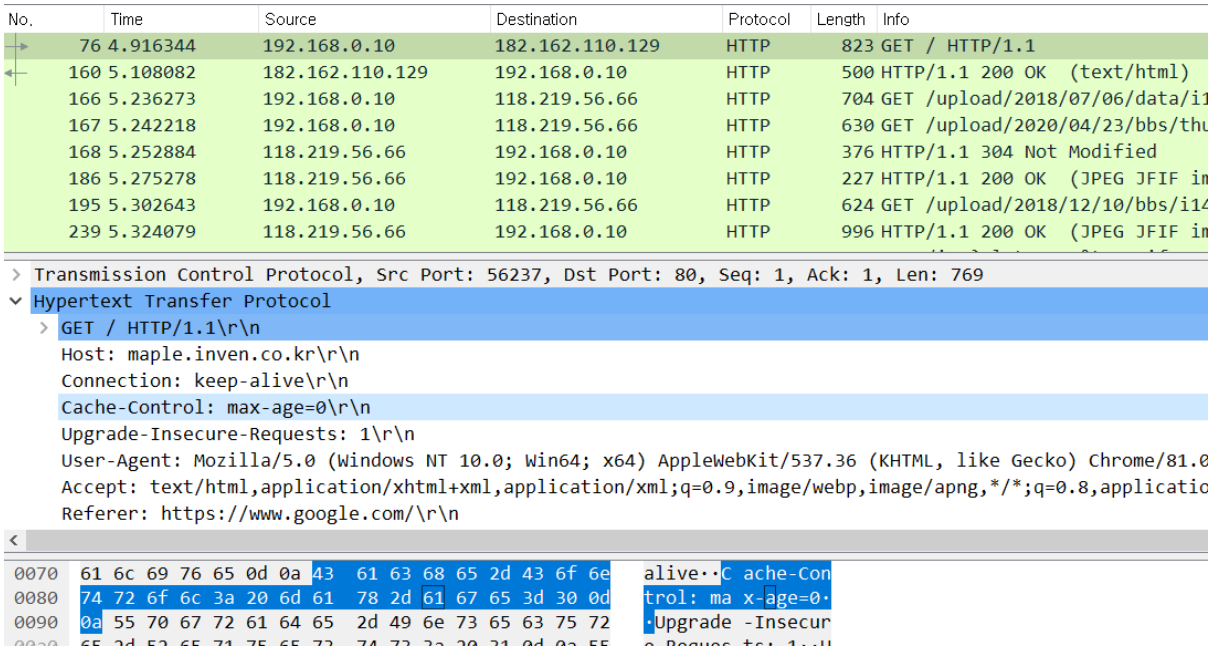
305 Use Proxy [RFC7231, Section 6.4.5]

306 (Unused) [RFC7231, Section 6.4.6]

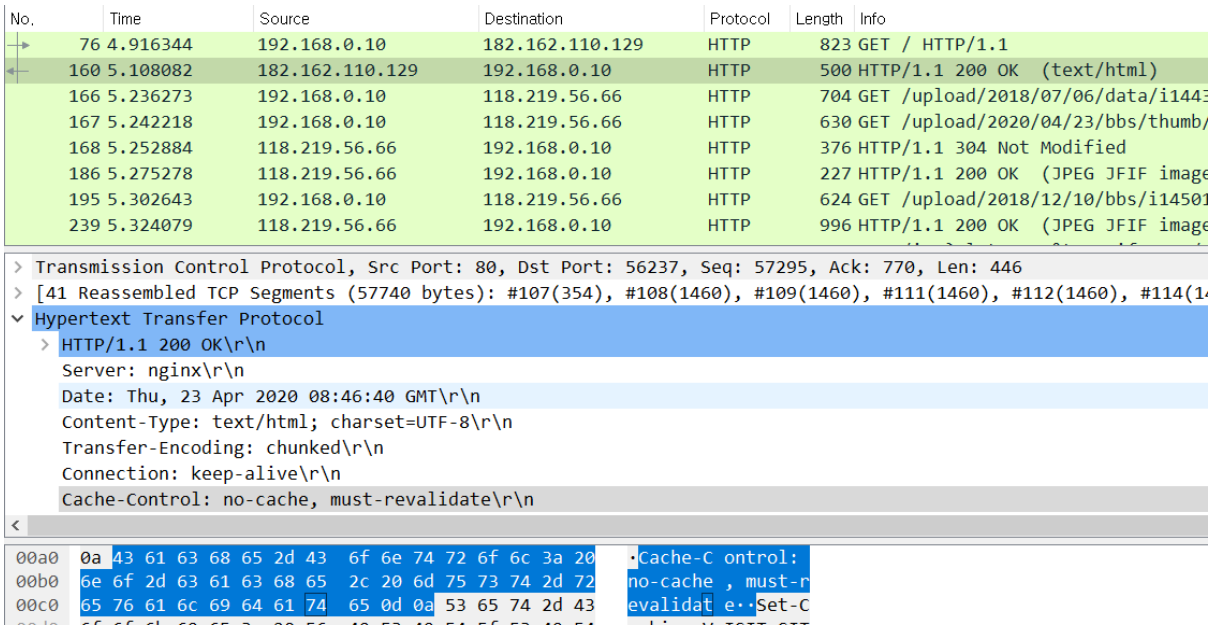
307 Temporary Redirect [RFC7231, Section 6.4.7]

308 Permanent Redirect [RFC7538]

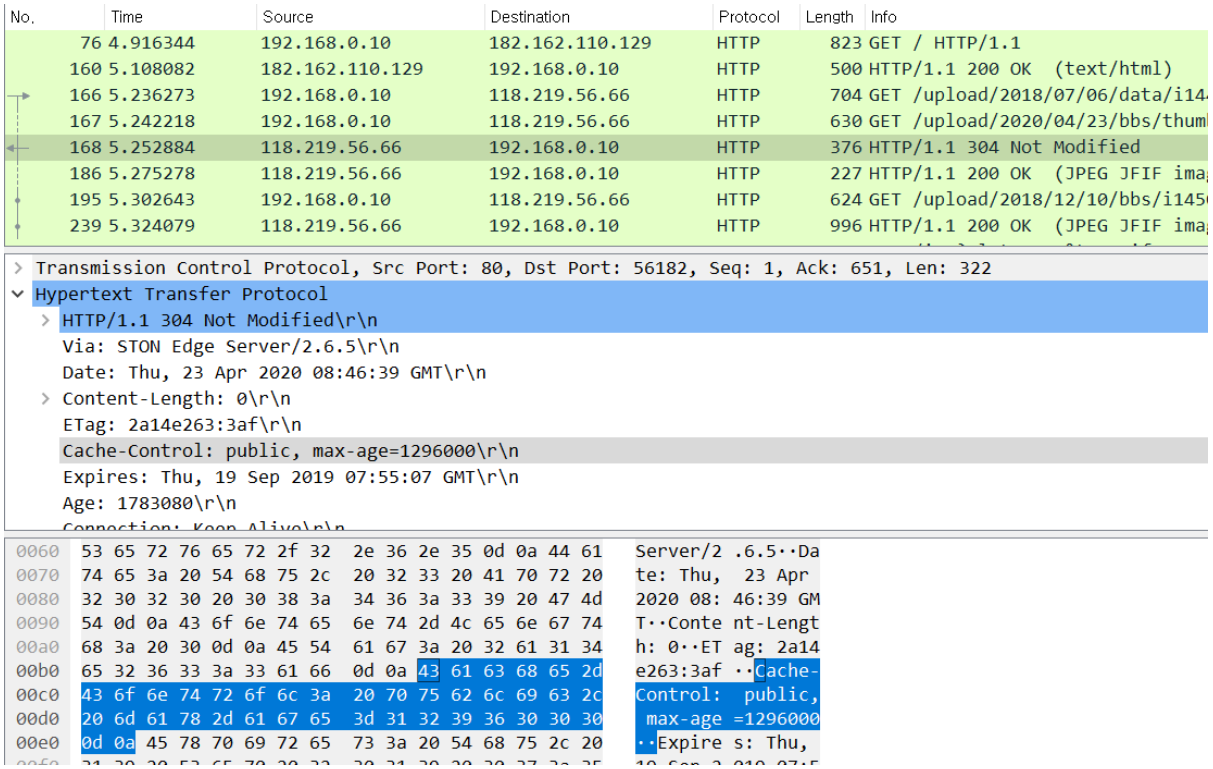
4) Web caching 기능



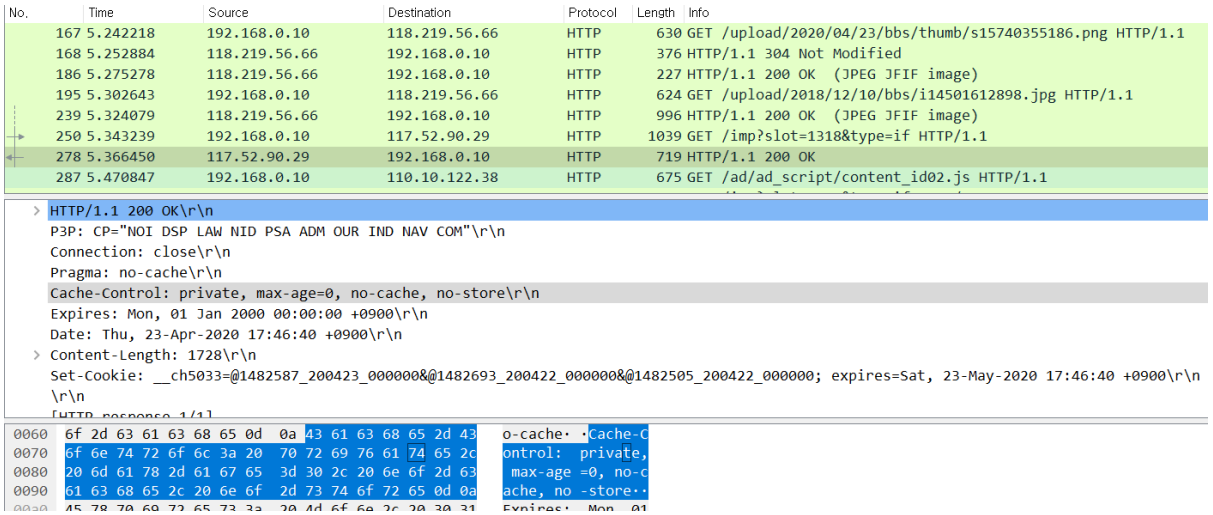
1. Max-age : 캐시의 유효시간을 나타낸다.



2. No-cache : 모든 캐시를 쓰기 전에 서버에 해당 캐시를 사용해도 되는지 확인하겠다.
3. Must-revalidate : 만료된 캐시만 서버에 확인하겠다.



4. Public : 공유 캐시에 저장해도 된다.
5. ETag : 캐시 업데이트 정보를 위한 임의의 식별 숫자이다.



6. Private : '브라우저'같은 특정 사용자 환경에만 저장한다.
7. No-store : 캐시를 저장하지 않는다.

5) 자유 주제

HTTP 와 HTTPS 의 차이점 : 암호화

HTTP 로만 된 사이트에서 로그인을 했을 때 패킷을 분석해 보면 암호화가 되어있지 않다. 암호화가 중요한 이유는 패킷이 중간에 스푸핑 될 수 있기 때문이다. 만약 해커가

로그인 할 때의 패킷을 가지면 username 과 password 를 알 수 있다. 실습한 사이트에서는 그 사이트에서 직접 로그인이 불가능하고 네이버 로그인 페이지로 넘어갔기 때문에 해당 내용을 확인할 수 없었다.

3. 새로 알게 된 지식

DNT(Do Not Track) : HTTP 요청 헤더 중 하나로 쿠키 등을 이용한 정보 추적을 금지한다는 의미를 웹서버에 전달한다.

각 프로토콜의 헤더만 분석하면 전송되는 데 필요한 정보들을 모두 얻을 수 있다.