

# Assignment2: Web Server Imitation

2017313008 김태은

## 1. Development environment

- Version of operating system: Ubuntu 18.04.4 LTS
- Programming language: Python
- Interpreter version: Python 3.6.9

server 실행 command : `python3 2017313008.py`

## 2. How to design

```
def main():  
  
    serverSocket=socket(AF_INET, SOCK_STREAM)  
    serverSocket.bind(('', 10080))  
    serverSocket.listen(100)  
    #print('The TCP server is ready to receive.\n')  
  
    while True:  
  
        connectionSocket, addr=serverSocket.accept()  
  
        t=Thread(target=execute, args=(connectionSocket, ))  
        t.start()  
  
    serverSocket.close()
```

**main 함수**

먼저 통신에 이용할 소켓을 생성하고, 포트(10080)에 맵핑(바인딩)한다. 이때 어떤 ip라도 모두 받을 수 있게 하기 위해 bind의 첫번째 인자를 empty형태로 넘긴다. 그리고 최대 100개의 연결 요청을 pending 하기 위해 listen(100)으로 설정했다. 서버는 concurrent HTTP request message를 처리할 수 있어야 하기 때문에 multi-thread 방식으로 구현하였다. 연결 요청 대기 중이던 serverSocket은 accept()하여 연결이 성립되고 connectionSocket, addr을 반환한다.

쓰레드에서 실행할 함수: execute, 그 함수에 넘겨질 인자: connectionSocket

html은 총 6개 생성하였다. index.html, secret.html, presecret.html, cookie.html, 403.html, 404.html

- index.html: 로그인 창 띄워주는 페이지
- secret.html: secret story 페이지
- presecret.html: index.html에서 넘겨받은 인자로 쿠키 설정하고, secret.html 띄워주는 페이지.
- cookie.html: user id과 쿠키가 만료될 때까지 남은 시간을 띄워주는 페이지
- 403.html: 403 Forbidden 띄워주는 페이지
- 404.html: 404 Not Found 띄워주는 페이지

```
def execute(connectionSocket):  
  
    count=100  
    while count>0: # Keep-alive: max=100  
  
        start_time=0.0 # cookie creation time (login post)  
        userid="" # userid in cookie  
        vTime=31 # valid time (time since cookie creation)  
  
        try:  
            msg=connectionSocket.recv(1024).decode()  
  
            if not msg:  
                connectionSocket.close()  
                break  
  
            #print('received: '+msg)  
  
            reset=0  
            if "userid" in msg and "passwd" in msg and "login" in msg:  
                userid=msg.split('\nuserid=')[1]  
                userid=userid.split('&')[0]  
                start_time=time.time()  
                vTime=time.time()-start_time  
                reset=1  
            elif '\nCookie: ' in msg:  
                c=msg.split('\nCookie: ')[1]  
                c=c.split('\n')[0]  
                start_time=float(c.split('=')[1])  
                vTime=time.time()-start_time
```

**execute 함수 (각 쓰레드마다 병렬적으로 실행됨)**

client가 보낸 request message는 msg에 저장된다. 만약 keep-alive: timeout=5 가 지나서 메시지가 들어오지 않으면 소켓을 종료한다. 또한 execute 함수안의 while count>0: 를 통해

keep-alive: max=100가 넘으면 소켓을 종료한다. 이것은 소켓이 한 번 사용 후 바로 종료되지 않고 재사용될 수 있게 소켓 close시기를 늦추는 것이다. (persistent http)

```
count=count-1
if count==0: connectionSocket.close()
```

while body 가장 아래에 존재

위의 1번: msg에 userid, passwd, login이 들어있을 때는 로그인 직후 presecret.html이 불렸을 때이다. msg에 나온 userid를 설정해주고, cookie가 생성된 시간인 start\_time을 현재 시간으로 설정해준다. vTime은 cookie 설정 후 얼마나 지났는지를 나타내므로 time.time()-start\_time으로 계산한다. 또한 presecret.html이 불렸을 때만 reponse header에 set-cookie를 넣어주기 위해 reset 변수를 하나 설정해둔다. (밑에서 쓰임)

위의 2번: 쿠키는 30초마다 만료되도록 밑에서 설정할 것이므로 만약 msg에 Cookie: 가 존재한다는 것은 쿠키 만료가 아직 안되었다는 뜻이다. (아직 생성 후 30초가 지나지 않음) 이 때는 쿠키에 저장되어 있는 start\_time을 읽어와서 vTime을 설정한다.

```
if msg.split()[1]=='/':
    fname='index.html'
else:
    filename=msg.split()[1]
    fname=filename[1:]

if vTime>=30:
    if fname!='index.html': raise ValueError

if fname=='index.html' and vTime<30:
    fname='secret.html'

f=open(fname, 'rb')
data=f.read()
f.close()

ext=fname.split('.')[1]
header=''
```

만약 client가 <http://serverIP:10080>로 접속된다면, request message가 GET / HTTP/1.1 와 같은 형태이므로 msg.split()[1]=='/'가 된다. 이 경우 index.html을 fname으로 지정한다.

만약 그렇지 않고 뒤에 html이나 img 파일 이름을 붙여서 접속된다면, fname을 그 파일이름으로 저장한다.

만약 `vTime >= 30`, 즉 쿠키 설정 이후 30초가 지났다면 `fname`이 `index.html`이 아닌 경우에 403 Forbidden을 띄워줘야 한다. 따라서 `valueError`을 `raise`해서 밑에서 따로 처리해준다.

만약 `fname`이 `index.html`이고 `vTime < 30`인 경우 `secret.html` 을 띄워줘야 하기 때문에 `fname`을 `secret.html`로 변경한다.

그렇게 `fname` 설정이 모두 끝나면 해당 파일을 오픈해서 읽어들이어 `data`에 저장한다.

또한 .을 뺀 파일의 확장자인 `ext`를 구한다.

```
if reset==1:
    header='HTTP/1.1 200 OK\r\nConnection: Keep-Alive\r\nKeep-Alive: timeout=5, max=%d\r\nContent-Length: %d\r\nSet-Cookie: %s=%d; Max-Age=30\r\nContent-Type: text/html\r\n\r\n'%(count, len(data), userid, start_time)

elif ext=='html':
    header='HTTP/1.1 200 OK\r\nConnection: Keep-Alive\r\nKeep-Alive: timeout=5, max=%d\r\nContent-Length: %d\r\nContent-Type: text/html\r\n\r\n'%(count, len(data))

else:
    if ext=='jpg' or ext=='jpe': ext='jpeg'
    mime='image/'+ext
    header='HTTP/1.1 200 OK\r\nConnection: Keep-Alive\r\nKeep-Alive: timeout=5, max=%d\r\nContent-Length: %d\r\nContent-Type: %s\r\n\r\n'%(count, len(data), mime)

connectionSocket.send(header.encode()+data)

#print('sent: '+header+'\n')
```

위에서 설명하였듯 로그인 후 `presecret.html` 호출이 들어오면 `reset=1`이 되므로, 이 경우에만 **Set-Cookie**를 통해 쿠키를 설정한다. Cookie에서 `name`은 `userid`, `value`는 `start_time`으로 설정하여, 이후 `cookie.html`에서 남은 시간을 계산할 수 있게 한다. Persistent http mode를 구현하기 위해 **keep-alive**로 설정했다. 또한 **keep-alive**가 유지되는 조건 또한 설정하였다. (`timeout=5, max=100`) 마지막으로 이 경우에 `html`이므로 `content-type`을 **text/html**로 설정한다.

만약 그 외에 확장자가 `html`인 파일 호출인 경우, 위의 `header`에서 **Set-Cookie**만 빼주면 동일하다.

그 외에 확장자가 `image`파일(`gif, jpg, jpeg, png` 등등)인 경우를 `else`로 묶어준다. `jpg, jpe, jpeg`의 `mime type`은 `image/jpeg`로 정해져 있어서 따로 처리해주었다. 여기서도 위의 `header`에서 **Set-Cookie**만 빼주면 동일하다.

이렇게 `header` 설정이 끝난 후 아까 저장한 `data`와 함께 `client`로 **response message**를 보낸다.

```

except ValueError:

    f=open('403.html', 'rb')
    data=f.read()
    f.close()

    header="HTTP/1.1 403 Forbidden\r\nConnection: Keep-Alive\r\nKeep-Alive: timeout=5, max=%d\r\nContent-Length: %d\r\nContent-Type: text/html\r\n\r\n"%(count, len(data))

    connectionSocket.send(header.encode()+data)

    #print('sent: '+header+'\n')

except IOError:

    f=open('404.html', 'rb')
    data=f.read()
    f.close()

    header="HTTP/1.1 404 Not Found\r\nConnection: Keep-Alive\r\nKeep-Alive: timeout=5, max=%d\r\nContent-Length: %d\r\nContent-Type: text/html\r\n\r\n"%(count, len(data))

    connectionSocket.send(header.encode()+data)

    #print('sent: '+header+'\n')

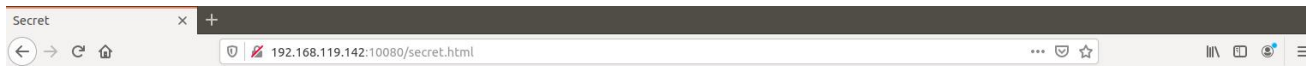
```

valueError의 경우 위에서 설명하였듯 따로 처리하기 위해 raise했다. 로그인 전에 index.html 외에는 아무 페이지도 접근하지 못하고 **403 Forbidden**을 출력해주어야 한다. 미리 만들어둔 403.html을 읽어들이고 아까와 똑같이 header 설정을 한다. header 설정이 끝나면 data와 함께 client에 response message를 보낸다.

IOError의 경우 로그인 후에 페이지에 접근하는데 해당 파일이 존재하지 않는 파일이어서 open이 되지 않을 때 발생한다. 이 경우에는 **404 Not Found**를 출력해주어야 한다. 미리 만들어둔 404.html을 읽어들이고 아까와 똑같이 header 설정을 한다. Header 설정이 끝나면 data와 함께 client에 response message를 보낸다.

### 3. functionalities

#### ① Basic HTTP requests & responses



#### My Travel Story

> 가장 기억에 남았던 여행을 소개합니다 <

(2019.02) 친구들과 함께 갔던 3박 4일 대만여행! 음식이 입맛에 잘 맞았고 골목, 시장 분위기가 엄청 마음에 들었다! 교통수단도 편리하고 택시투어 프로그램도 잘 되어 있어서 편하게 여행할 수 있었다. 비가 오는 곳은 날씨였지만 사람들이 친절해서 좋은 기억으로 남아있다. 다음에 꼭 다시 가보고 싶은 여행지! (다음엔 가오슝에 가볼 것이다)



여러 그림이 들어있는 html을 요청하면, 브라우저가 추가적으로 그림에 대한 request message를 서버로 보낸다.

```
received: GET /taiwan-1.jpg HTTP/1.1
Host: 192.168.119.142:10080
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:75.0) Gecko/20100101 Firefox/75.0
Accept: image/webp,*/*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Referer: http://192.168.119.142:10080/secret.html
Cookie: taeeun=1587240713

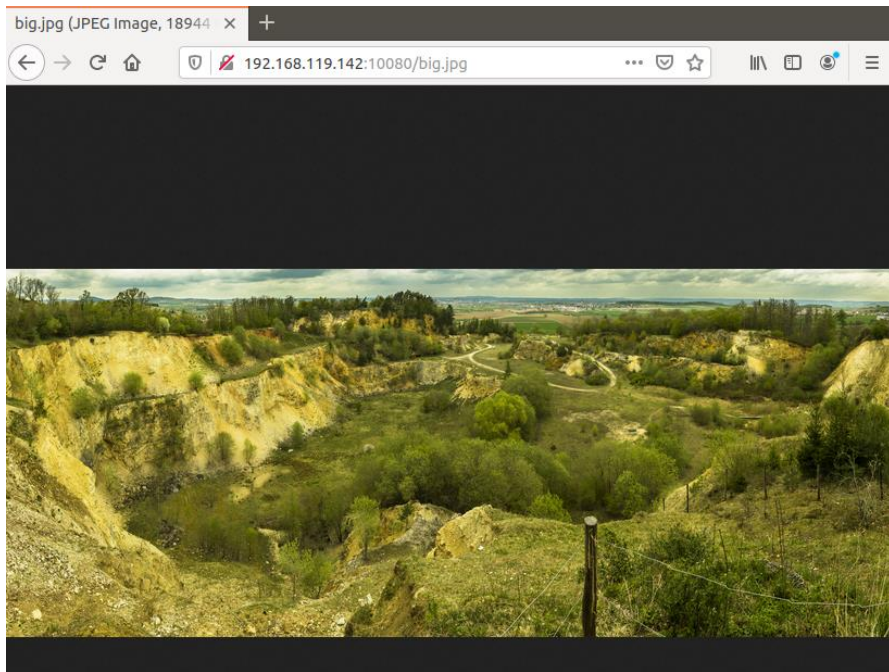
received: GET /taiwan-2.jpg HTTP/1.1
Host: 192.168.119.142:10080
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:75.0) Gecko/20100101 Firefox/75.0
Accept: image/webp,*/*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Referer: http://192.168.119.142:10080/secret.html
Cookie: taeeun=1587240713

received: GET /taiwan-3.jpg HTTP/1.1
Host: 192.168.119.142:10080
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:75.0) Gecko/20100101 Firefox/75.0
Accept: image/webp,*/*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Referer: http://192.168.119.142:10080/secret.html
Cookie: taeeun=1587240713

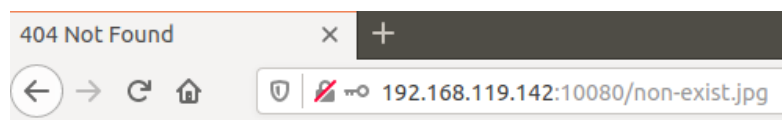
received: GET /taiwan-4.jpg HTTP/1.1
Host: 192.168.119.142:10080
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:75.0) Gecko/20100101 Firefox/75.0
Accept: image/webp,*/*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Referer: http://192.168.119.142:10080/secret.html
Cookie: taeeun=1587240713
```

secret.html에 4개의 이미지가 있기 때문에 4번의 request가 더 온 것을 확인할 수 있다.





51.5MB의 용량이 비교적 큰 image도 잘 전송된다.

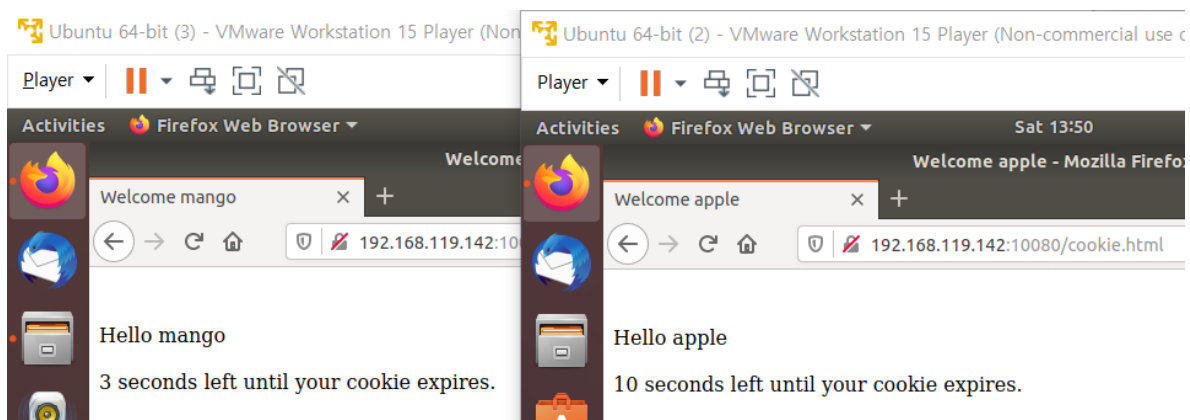


## 404 Not Found

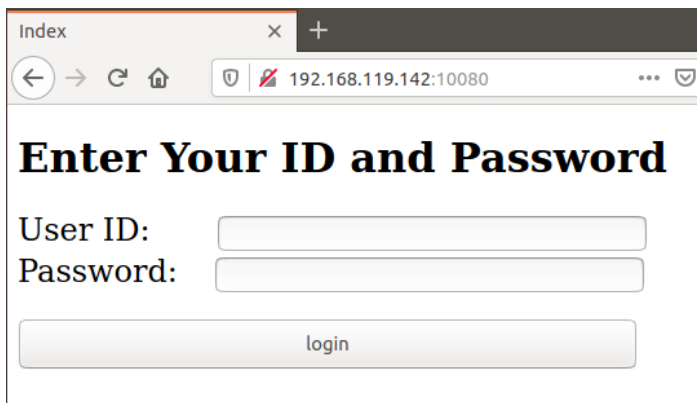
서버를 실행하는 컴퓨터에 없는 이미지를 요청하면 404 Not Found가 출력된다.

멀티 쓰레드 방식으로 구현하였기 때문에 concurrent user가 반복적으로 content를 요청하는 것을 수용할 수 있다.

만약 여러 대의 컴퓨터에서 동시에 요청이 들어와도 수용 가능하다.



## ② Log-in functionality



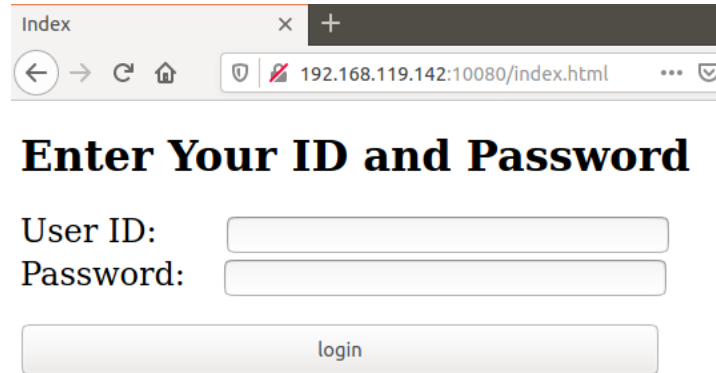
Index

Enter Your ID and Password

User ID:

Password:

login



Index

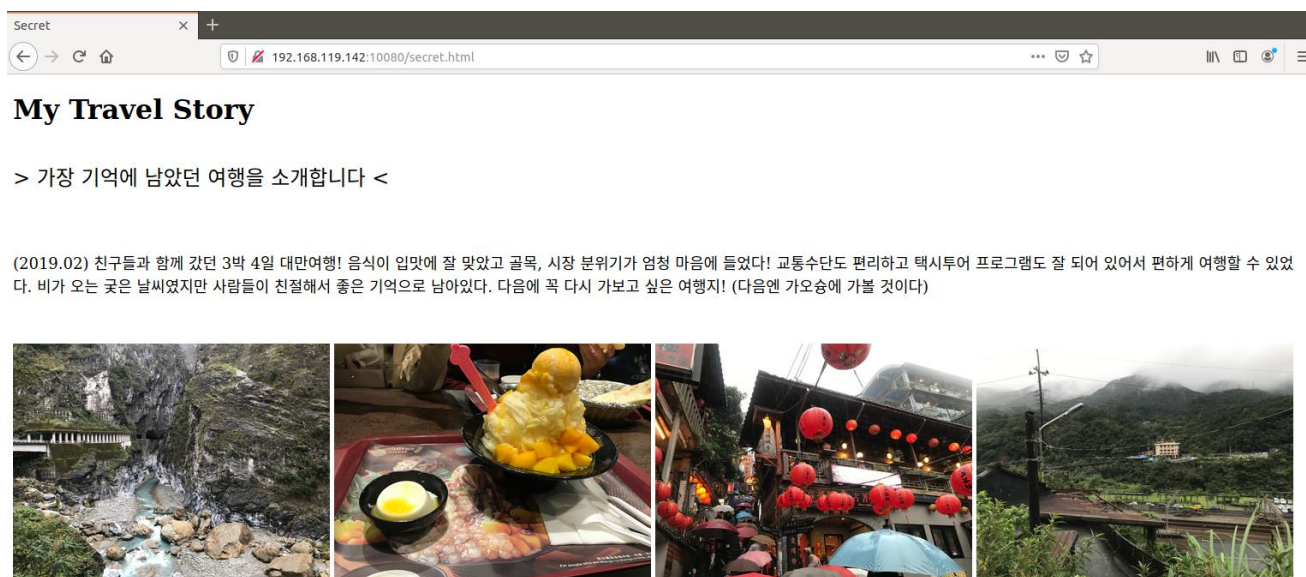
Enter Your ID and Password

User ID:

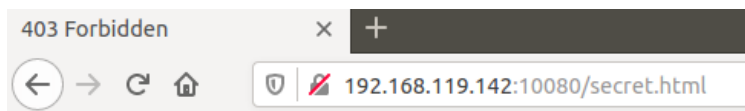
Password:

login

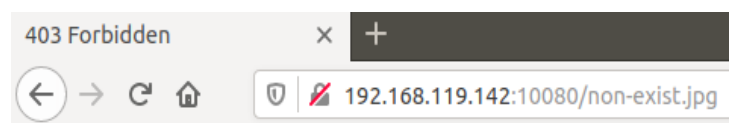
http://serverIP:10080 또는 http://serverIP:10080/index.html 접속해서 index.html 접근



로그인 후 secret.html 접근



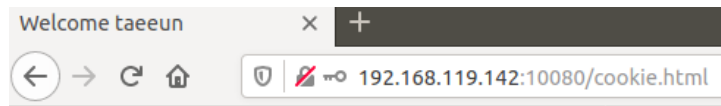
## 403 Forbidden



## 403 Forbidden

만약 로그인 하지 않았거나, 쿠키가 만료되었을 때 secret.html을 접근하면 403 Forbidden이 뜨는 것을 확인할 수 있다. 또한 존재하지 않는 파일에 대해서도 동일한 메시지를 띄운다.





Hello taeun

26 seconds left until your cookie expires.

쿠키가 만료되기 전에 cookie.html에 접속하면 user\_id와 쿠키 만료까지 남은 시간을 볼 수 있다.

### ③ Persistent HTTP mode

wireshark를 통해 확인해보면,

283	30.573868	192.168.119.141	192.168.119.142	HTTP	498	GET /secret.html HTTP/1.1
285	30.574445	192.168.119.142	192.168.119.141	HTTP	1163	HTTP/1.1 200 OK (text/html)
287	30.676021	192.168.119.141	192.168.119.142	HTTP	406	GET /taiwan-1.jpg HTTP/1.1
292	30.685583	192.168.119.141	192.168.119.142	HTTP	406	GET /taiwan-2.jpg HTTP/1.1
297	30.691342	192.168.119.141	192.168.119.142	HTTP	406	GET /taiwan-3.jpg HTTP/1.1
302	30.696123	192.168.119.141	192.168.119.142	HTTP	406	GET /taiwan-4.jpg HTTP/1.1
5108	30.821269	192.168.119.142	192.168.119.141	HTTP	111	HTTP/1.1 200 OK (JPEG JFIF image)
6663	30.883232	192.168.119.142	192.168.119.141	HTTP	1243	HTTP/1.1 200 OK (JPEG JFIF image)
8256	30.967830	192.168.119.142	192.168.119.141	HTTP	1225	HTTP/1.1 200 OK (JPEG JFIF image)
9586	31.004527	192.168.119.142	192.168.119.141	HTTP	1331	HTTP/1.1 200 OK (JPEG JFIF image)
9588	32.342074	192.168.119.141	192.168.119.142	HTTP	470	GET /cookie.html HTTP/1.1
9589	32.343067	192.168.119.142	192.168.119.141	HTTP	815	HTTP/1.1 200 OK (text/html)
9591	33.186601	192.168.119.141	192.168.119.142	HTTP	524	GET /secret.html HTTP/1.1
9592	33.186954	192.168.119.142	192.168.119.141	HTTP	1163	HTTP/1.1 200 OK (text/html)
9594	33.267599	192.168.119.141	192.168.119.142	HTTP	432	GET /taiwan-1.jpg HTTP/1.1
10946	33.315120	192.168.119.141	192.168.119.142	HTTP	432	GET /taiwan-2.jpg HTTP/1.1
11686	33.342651	192.168.119.141	192.168.119.142	HTTP	432	GET /taiwan-3.jpg HTTP/1.1
13336	33.378450	192.168.119.141	192.168.119.142	HTTP	432	GET /taiwan-4.jpg HTTP/1.1
17906	33.512657	192.168.119.142	192.168.119.141	HTTP	1139	HTTP/1.1 200 OK (JPEG JFIF image)
18094	33.517389	192.168.119.142	192.168.119.141	HTTP	110	HTTP/1.1 200 OK (JPEG JFIF image)
18356	33.542173	192.168.119.142	192.168.119.141	HTTP	1224	HTTP/1.1 200 OK (JPEG JFIF image)

Frame 283: 498 bytes on wire (3984 bits), 498 bytes captured (3984 bits) on interface \Device\NPF\_{74B4B6FC-22A2-4...  
Ethernet II, Src: VMware\_f5:b6:87 (00:0c:29:f5:b6:87), Dst: VMware\_60:41:a2 (00:0c:29:60:41:a2)  
Internet Protocol Version 4, Src: 192.168.119.141, Dst: 192.168.119.142  
Transmission Control Protocol, Src Port: 50584, Dst Port: 10080, Seq: 891, Ack: 575, Len: 432  
Source Port: 50584  
Destination Port: 10080

No.	Time	Source	Destination	Protocol	Length	Info
283	30.573868	192.168.119.141	192.168.119.142	HTTP	498	GET /secret.html HTTP/1.1
285	30.574445	192.168.119.142	192.168.119.141	HTTP	1163	HTTP/1.1 200 OK (text/html)
287	30.676021	192.168.119.141	192.168.119.142	HTTP	406	GET /taiwan-1.jpg HTTP/1.1
292	30.685583	192.168.119.141	192.168.119.142	HTTP	406	GET /taiwan-2.jpg HTTP/1.1
297	30.691342	192.168.119.141	192.168.119.142	HTTP	406	GET /taiwan-3.jpg HTTP/1.1
302	30.696123	192.168.119.141	192.168.119.142	HTTP	406	GET /taiwan-4.jpg HTTP/1.1
5108	30.821269	192.168.119.142	192.168.119.141	HTTP	111	HTTP/1.1 200 OK (JPEG JFIF image)
6663	30.883232	192.168.119.142	192.168.119.141	HTTP	1243	HTTP/1.1 200 OK (JPEG JFIF image)
8256	30.967830	192.168.119.142	192.168.119.141	HTTP	1225	HTTP/1.1 200 OK (JPEG JFIF image)
9586	31.004527	192.168.119.142	192.168.119.141	HTTP	1331	HTTP/1.1 200 OK (JPEG JFIF image)
9588	32.342074	192.168.119.141	192.168.119.142	HTTP	470	GET /cookie.html HTTP/1.1
9589	32.343067	192.168.119.142	192.168.119.141	HTTP	815	HTTP/1.1 200 OK (text/html)
9591	33.186601	192.168.119.141	192.168.119.142	HTTP	524	GET /secret.html HTTP/1.1
9592	33.186954	192.168.119.142	192.168.119.141	HTTP	1163	HTTP/1.1 200 OK (text/html)
9594	33.267599	192.168.119.141	192.168.119.142	HTTP	432	GET /taiwan-1.jpg HTTP/1.1
10946	33.315120	192.168.119.141	192.168.119.142	HTTP	432	GET /taiwan-2.jpg HTTP/1.1
11686	33.342651	192.168.119.141	192.168.119.142	HTTP	432	GET /taiwan-3.jpg HTTP/1.1
13336	33.378450	192.168.119.141	192.168.119.142	HTTP	432	GET /taiwan-4.jpg HTTP/1.1
17906	33.512657	192.168.119.142	192.168.119.141	HTTP	1139	HTTP/1.1 200 OK (JPEG JFIF image)
18094	33.517389	192.168.119.142	192.168.119.141	HTTP	110	HTTP/1.1 200 OK (JPEG JFIF image)
18356	33.542173	192.168.119.142	192.168.119.141	HTTP	1224	HTTP/1.1 200 OK (JPEG JFIF image)

> Frame 287: 406 bytes on wire (3248 bits), 406 bytes captured (3248 bits) on interface \Device\NPF\_{74B4B6FC-22A2-4000-8000-000000000000}

> Ethernet II, Src: VMware\_f5:b6:87 (00:0c:29:f5:b6:87), Dst: VMware\_60:41:a2 (00:0c:29:60:41:a2)

> Internet Protocol Version 4, Src: 192.168.119.141, Dst: 192.168.119.142

> Transmission Control Protocol, Src Port: 50584, Dst Port: 10080, Seq: 1323, Ack: 1672, Len: 340

Source Port: 50584

Destination Port: 10080

No.	Time	Source	Destination	Protocol	Length	Info
283	30.573868	192.168.119.141	192.168.119.142	HTTP	498	GET /secret.html HTTP/1.1
285	30.574445	192.168.119.142	192.168.119.141	HTTP	1163	HTTP/1.1 200 OK (text/html)
287	30.676021	192.168.119.141	192.168.119.142	HTTP	406	GET /taiwan-1.jpg HTTP/1.1
292	30.685583	192.168.119.141	192.168.119.142	HTTP	406	GET /taiwan-2.jpg HTTP/1.1
297	30.691342	192.168.119.141	192.168.119.142	HTTP	406	GET /taiwan-3.jpg HTTP/1.1
302	30.696123	192.168.119.141	192.168.119.142	HTTP	406	GET /taiwan-4.jpg HTTP/1.1
5108	30.821269	192.168.119.142	192.168.119.141	HTTP	111	HTTP/1.1 200 OK (JPEG JFIF image)
6663	30.883232	192.168.119.142	192.168.119.141	HTTP	1243	HTTP/1.1 200 OK (JPEG JFIF image)
8256	30.967830	192.168.119.142	192.168.119.141	HTTP	1225	HTTP/1.1 200 OK (JPEG JFIF image)
9586	31.004527	192.168.119.142	192.168.119.141	HTTP	1331	HTTP/1.1 200 OK (JPEG JFIF image)
9588	32.342074	192.168.119.141	192.168.119.142	HTTP	470	GET /cookie.html HTTP/1.1
9589	32.343067	192.168.119.142	192.168.119.141	HTTP	815	HTTP/1.1 200 OK (text/html)
9591	33.186601	192.168.119.141	192.168.119.142	HTTP	524	GET /secret.html HTTP/1.1
9592	33.186954	192.168.119.142	192.168.119.141	HTTP	1163	HTTP/1.1 200 OK (text/html)
9594	33.267599	192.168.119.141	192.168.119.142	HTTP	432	GET /taiwan-1.jpg HTTP/1.1
10946	33.315120	192.168.119.141	192.168.119.142	HTTP	432	GET /taiwan-2.jpg HTTP/1.1
11686	33.342651	192.168.119.141	192.168.119.142	HTTP	432	GET /taiwan-3.jpg HTTP/1.1
13336	33.378450	192.168.119.141	192.168.119.142	HTTP	432	GET /taiwan-4.jpg HTTP/1.1
17906	33.512657	192.168.119.142	192.168.119.141	HTTP	1139	HTTP/1.1 200 OK (JPEG JFIF image)
18094	33.517389	192.168.119.142	192.168.119.141	HTTP	110	HTTP/1.1 200 OK (JPEG JFIF image)
18356	33.542173	192.168.119.142	192.168.119.141	HTTP	1224	HTTP/1.1 200 OK (JPEG JFIF image)

> Frame 9588: 470 bytes on wire (3760 bits), 470 bytes captured (3760 bits) on interface \Device\NPF\_{74B4B6FC-22A2-4000-8000-000000000000}

> Ethernet II, Src: VMware\_f5:b6:87 (00:0c:29:f5:b6:87), Dst: VMware\_60:41:a2 (00:0c:29:60:41:a2)

> Internet Protocol Version 4, Src: 192.168.119.141, Dst: 192.168.119.142

> Transmission Control Protocol, Src Port: 50584, Dst Port: 10080, Seq: 1663, Ack: 4947665, Len: 404

Source Port: 50584

Destination Port: 10080

No.	Time	Source	Destination	Protocol	Length	Info
283	30.573868	192.168.119.141	192.168.119.142	HTTP	498	GET /secret.html HTTP/1.1
285	30.574445	192.168.119.142	192.168.119.141	HTTP	1163	HTTP/1.1 200 OK (text/html)
287	30.676021	192.168.119.141	192.168.119.142	HTTP	406	GET /taiwan-1.jpg HTTP/1.1
292	30.685583	192.168.119.141	192.168.119.142	HTTP	406	GET /taiwan-2.jpg HTTP/1.1
297	30.691342	192.168.119.141	192.168.119.142	HTTP	406	GET /taiwan-3.jpg HTTP/1.1
302	30.696123	192.168.119.141	192.168.119.142	HTTP	406	GET /taiwan-4.jpg HTTP/1.1
5108	30.821269	192.168.119.142	192.168.119.141	HTTP	111	HTTP/1.1 200 OK (JPEG JFIF image)
6663	30.883232	192.168.119.142	192.168.119.141	HTTP	1243	HTTP/1.1 200 OK (JPEG JFIF image)
8256	30.967830	192.168.119.142	192.168.119.141	HTTP	1225	HTTP/1.1 200 OK (JPEG JFIF image)
9586	31.004527	192.168.119.142	192.168.119.141	HTTP	1331	HTTP/1.1 200 OK (JPEG JFIF image)
9588	32.342074	192.168.119.141	192.168.119.142	HTTP	470	GET /cookie.html HTTP/1.1
9589	32.343067	192.168.119.142	192.168.119.141	HTTP	815	HTTP/1.1 200 OK (text/html)
9591	33.186601	192.168.119.141	192.168.119.142	HTTP	524	GET /secret.html HTTP/1.1
9592	33.186954	192.168.119.142	192.168.119.141	HTTP	1163	HTTP/1.1 200 OK (text/html)
9594	33.267599	192.168.119.141	192.168.119.142	HTTP	432	GET /taiwan-1.jpg HTTP/1.1
10946	33.315120	192.168.119.141	192.168.119.142	HTTP	432	GET /taiwan-2.jpg HTTP/1.1
11686	33.342651	192.168.119.141	192.168.119.142	HTTP	432	GET /taiwan-3.jpg HTTP/1.1
13336	33.378450	192.168.119.141	192.168.119.142	HTTP	432	GET /taiwan-4.jpg HTTP/1.1
17906	33.512657	192.168.119.142	192.168.119.141	HTTP	1139	HTTP/1.1 200 OK (JPEG JFIF image)
18094	33.517389	192.168.119.142	192.168.119.141	HTTP	110	HTTP/1.1 200 OK (JPEG JFIF image)
18356	33.542173	192.168.119.142	192.168.119.141	HTTP	1224	HTTP/1.1 200 OK (JPEG JFIF image)
> Frame 9594: 432 bytes on wire (3456 bits), 432 bytes captured (3456 bits) on interface \Device\NPF_{74B4B6FC-22A > Ethernet II, Src: VMware_f5:b6:87 (00:0c:29:f5:b6:87), Dst: VMware_60:41:a2 (00:0c:29:60:41:a2) > Internet Protocol Version 4, Src: 192.168.119.141, Dst: 192.168.119.142 > Transmission Control Protocol, Src Port: 50584, Dst Port: 10080, Seq: 2525, Ack: 4949511, Len: 366 Source Port: 50584 Destination Port: 10080						

No.	Time	Source	Destination	Protocol	Length	Info
283	30.573868	192.168.119.141	192.168.119.142	HTTP	498	GET /secret.html HTTP/1.1
285	30.574445	192.168.119.142	192.168.119.141	HTTP	1163	HTTP/1.1 200 OK (text/html)
287	30.676021	192.168.119.141	192.168.119.142	HTTP	406	GET /taiwan-1.jpg HTTP/1.1
292	30.685583	192.168.119.141	192.168.119.142	HTTP	406	GET /taiwan-2.jpg HTTP/1.1
297	30.691342	192.168.119.141	192.168.119.142	HTTP	406	GET /taiwan-3.jpg HTTP/1.1
302	30.696123	192.168.119.141	192.168.119.142	HTTP	406	GET /taiwan-4.jpg HTTP/1.1
5108	30.821269	192.168.119.142	192.168.119.141	HTTP	111	HTTP/1.1 200 OK (JPEG JFIF image)
6663	30.883232	192.168.119.142	192.168.119.141	HTTP	1243	HTTP/1.1 200 OK (JPEG JFIF image)
8256	30.967830	192.168.119.142	192.168.119.141	HTTP	1225	HTTP/1.1 200 OK (JPEG JFIF image)
9586	31.004527	192.168.119.142	192.168.119.141	HTTP	1331	HTTP/1.1 200 OK (JPEG JFIF image)
9588	32.342074	192.168.119.141	192.168.119.142	HTTP	470	GET /cookie.html HTTP/1.1
9589	32.343067	192.168.119.142	192.168.119.141	HTTP	815	HTTP/1.1 200 OK (text/html)
9591	33.186601	192.168.119.141	192.168.119.142	HTTP	524	GET /secret.html HTTP/1.1
9592	33.186954	192.168.119.142	192.168.119.141	HTTP	1163	HTTP/1.1 200 OK (text/html)
9594	33.267599	192.168.119.141	192.168.119.142	HTTP	432	GET /taiwan-1.jpg HTTP/1.1
10946	33.315120	192.168.119.141	192.168.119.142	HTTP	432	GET /taiwan-2.jpg HTTP/1.1
11686	33.342651	192.168.119.141	192.168.119.142	HTTP	432	GET /taiwan-3.jpg HTTP/1.1
13336	33.378450	192.168.119.141	192.168.119.142	HTTP	432	GET /taiwan-4.jpg HTTP/1.1
17906	33.512657	192.168.119.142	192.168.119.141	HTTP	1139	HTTP/1.1 200 OK (JPEG JFIF image)
18094	33.517389	192.168.119.142	192.168.119.141	HTTP	110	HTTP/1.1 200 OK (JPEG JFIF image)
18356	33.542173	192.168.119.142	192.168.119.141	HTTP	1224	HTTP/1.1 200 OK (JPEG JFIF image)

Host: 192.168.119.142:10080\r\n  
 User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86\_64; rv:75.0) Gecko/20100101 Firefox/75.0\r\n  
 Accept: image/webp,\*/\*\r\n  
 Accept-Language: en-US,en;q=0.5\r\n  
 Accept-Encoding: gzip, deflate\r\n  
 Connection: keep-alive\r\n  
 Referer: http://192.168.119.142:10080/secret.html\r\n  
 > Cookie: taeeun=1587241321\r\n  
 Cache-Control: max-age=0\r\n  
 \r\n  
 [Full request URI: http://192.168.119.142:10080/taiwan-1.jpg]  
 [HTTP request 7/7]  
 [Prev request in frame: 9591]  
 [Response in frame: 17906]

하나의 TCP connection이 response message를 보낸 후 종료되지 않고 계속 **재사용**되고 있다.

여기서는 port number 50584를 예시로 보여주었지만 wireshark로 확인하면 다른 TCP connection(port number)도 계속 재사용되고 있음을 확인할 수 있다.

**(persistent HTTP)**