

# Computer Network HW 3

학번 / 이름: 2014310407 / 이 준 혁

## 1. 개발 환경

- OS: Windows 10
- Programming language: Python 3.7
- IDE: Pycharm 2018.1.4

## 2. 구현

### 2.1 구현 내용 요약

4.1. Basic Http requests & responses: 전부

4.2. Login functionality:

- 1) 서버의 첫 페이지 접근 시 index.html 을 전송 & index.html 제작
- 2) URL 에 직접 secret.html 접속 시 403 Forbidden 에러 메시지 출력
- 3) secret.html 제작 및 secret.html 을 index.html 의 log-in 을 통해 접근 가능하도록 설정

4.3. Persistent HTTP mode: 미 구현

### 2.2 구현 세부 설명

#### 2.2.1 server.py

가장 먼저 socket 패키지, 멀티스레드를 위한 threading 패키지, cookie 의 expire time 을 위한 datetime 패키지를 import 했다. 추가로 getsizeof 함수를 가져왔다.

전역변수로는 Buffersize 를 1024 로 고정했고, serverport 는 10080 으로, http message 를 위한 WrWn 은 CRLF 로 정의했다.

메인 함수가 실행되면 가장 먼저 TCP 연결을 위한 socket 을 만들고 bind 와 listen 함수를 통해, 주어진 포트에 해당되는 연결을 최대 10 개까지 받을 수 있게(큐에 넣을 수 있게) 설정했다. 이후 무한루프를 통해 socket 이 TCP connection 을 accept 하면 serverConnect 함수를 실행하도록 했다. 이를 병렬적으로 수행하기 위해 threading 패키지의 Thread 함수를 통해 멀티스레드를 구현했다.

serverConnect 함수는 연결된 소켓을 인자로 받아서 실행되는 함수이다. 우선 client 로부터 읽어 들일 message 인 requestmessage 와 client 에게 보낼 message 인 responsemessage 를 선언한다. 그리고 무한루프를 통해 버퍼 크기만큼의 데이터를 입력으로 받는데, 만약 더 이상 받아들일 입력이 없거나 받아들인 데이터의 크기가 버퍼크기보다 작다면 파일이 끝이라는 의미이므로 반복문을 탈출한다. 이렇게 받아들인 값들을 차례로 requestmessage 에 저장한다.

이렇게 받아들인 requestmessage 를 CRLF 를 기준으로 split 한다. 만약 이렇게 split 을 했는데 값이 하나도 없다면 request 가 없었다는 뜻이므로 바로 socket 을 닫고 종료했다. requestmessage 가 있는 경우 메시지 파싱을 통해 method 와 request 한 filename 을 알아냈다. 만약 method 가 POST 혹은 GET 의 형태가 아닌 경우 responsemessage 를 HTTP/1.1 400 Bad Request 로 설정 후 client 에게 전송한 후 종료했다.

만약 Method 가 GET 혹은 POST 라면, filename 을 확인해서 다음과 같은 경우로 나누었다.

1. filename 이 없는 경우; 즉, [http://\\*\\*\\*.\\*\\*\\*.\\*\\*\\*.\\*\\*\\*:10080](http://***.***.***.***:10080) 으로 단순 접속한 경우 -> 이 경우는 index.html 을 client 에게 보내주어야 하므로 filename 을 index.html 로 바꾸어 [http://\\*\\*\\*.\\*\\*\\*.\\*\\*\\*.\\*\\*\\*:10080/index.html](http://***.***.***.***:10080/index.html) 을 입력한 경우와 같이 취급했다.

2. filename 이 secret.html 인 경우; requestmessage 중에서 "Referer"헤더가 있는지 판별했다. Referer 헤더가 있다면 정상적으로 접근한 것이고 없다면 URL 을 통해 직접 접근한 것이므로 HTTP/1.1 403 Forbidden 을 responsemessage 로 설정하고 이를 client 에게 전송 후 소켓을 닫고 종료했다.

3. filename 이 있지만, 해당 file 이 존재하지 않는 경우; HTTP/1.1 404 Not Found 을 responsemessage 로 설정하고 이를 client 에게 전송 후 소켓을 닫고 종료했다.

4. filename 이 있고 해당 file 이 존재하는 경우; 우선 HTTP/1.1 200 OK 를 response message 의 처음으로 둔다.

//이후 TCP connection + Cookie

이렇게 완성된 responsemessage 를 client 에게 보낸다. 이후 요청한 파일을 client 에게 보내야 하므로 read 함수를 통해 file 을 버퍼의 크기만큼 읽고 이렇게 읽은 값을 계속 client 에게 보내줬다. 이 모든 과정이 끝나면 소켓을 닫고 0 을 반환했다.

### 2.2.2 index.html

Enter your ID and Password

User ID :

PassWord :

login

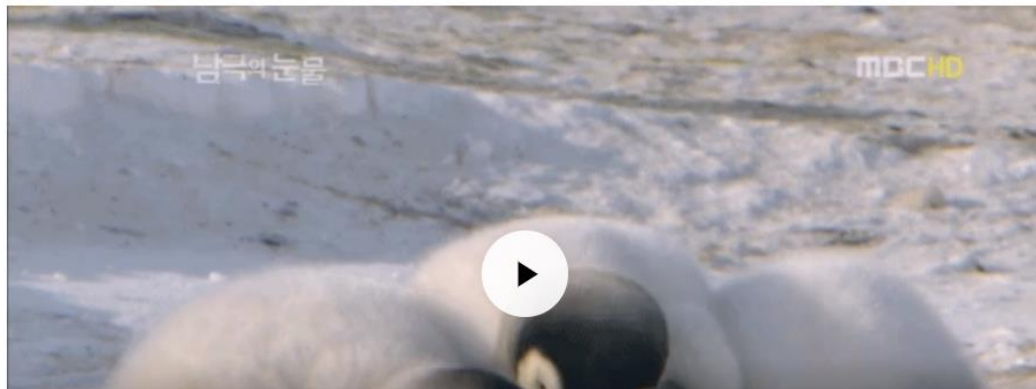
위와 같이 ID, PW 를 입력하는 값과 login 버튼을 만들었다. login 버튼을 누르면 secret.html 으로 이동되도록 했다.

### 2.2.3 secret.html



(bikeriderlondon/shutterstock.com)

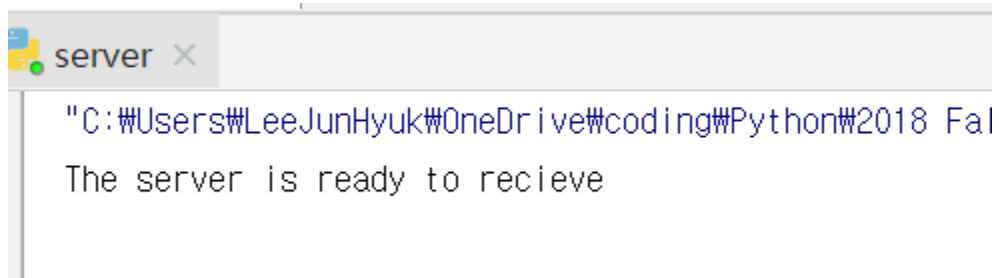
t legs, their legs are quite long. Totla leg size is almostly the half of their body!



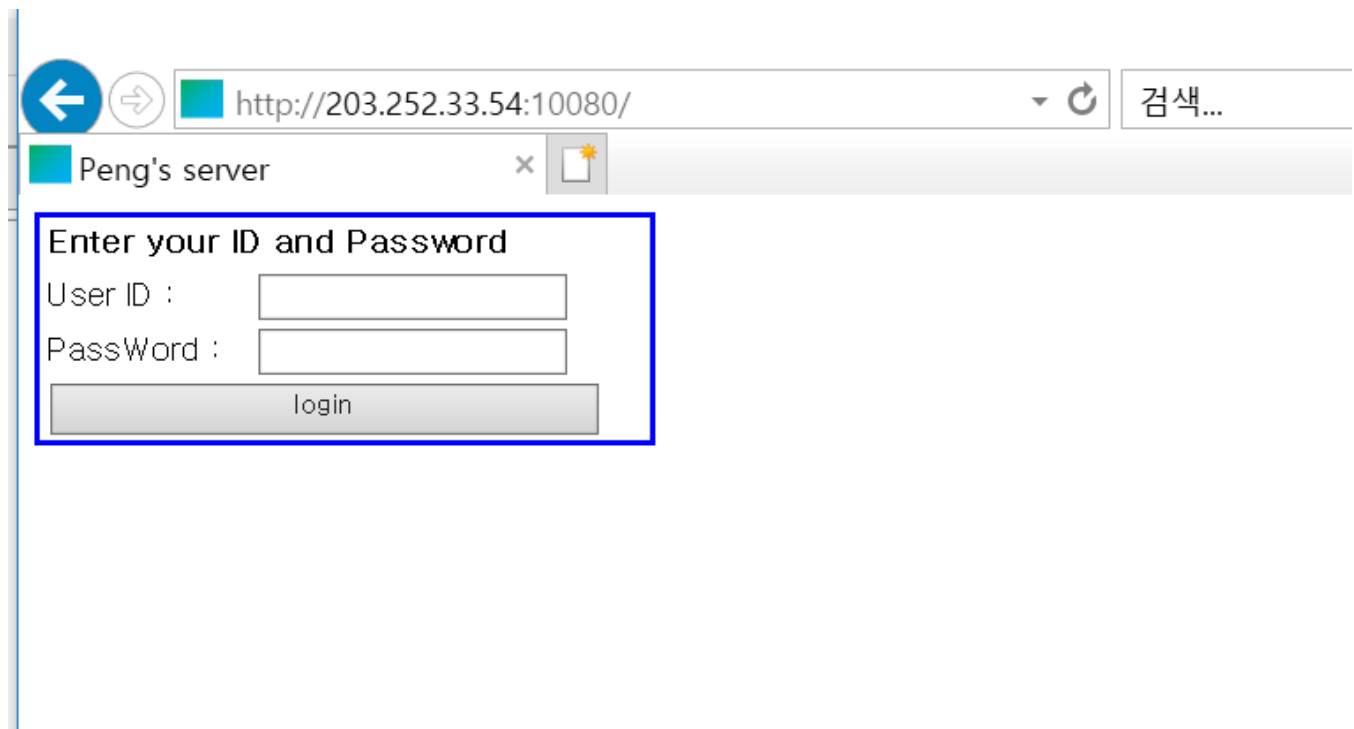
위와 같이 여러 이미지 파일, 동영상 등이 잘 보이도록 구성했다. (위 이미지는 일부 발췌한 것)

### 3. 결과

(여기에서는 favicon.ico 파일이 존재하므로 로고가 있다.)



- 서버실행

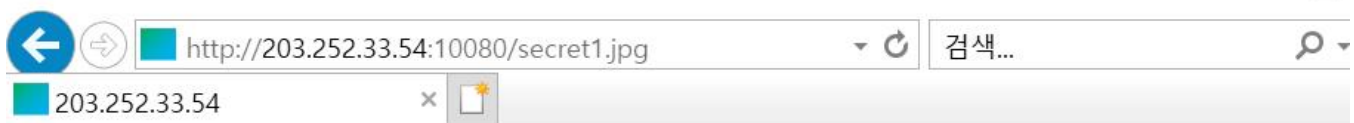


- IP 주소를 통해 접근. 잘 접근됨을 확인했다.

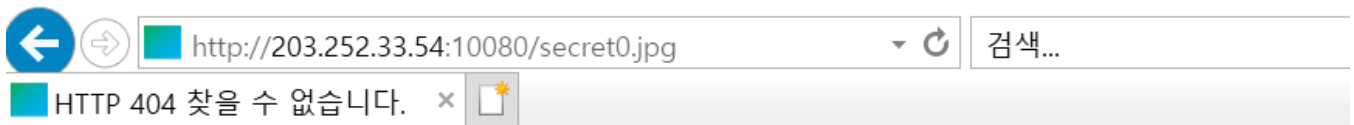
```
server x
"C:\Users\LeeJunHyuk\OneDrive\coding\Python\2018 Fall Network\server\venv\Scripts\python.exe" "C:/Users/LeeJunHyuk/OneDrive/coding/Python/2018 Fall Network/server/venv/Scripts/python.exe"
The server is ready to recieve
GET / HTTP/1.1
Accept: text/html, application/xhtml+xml, image/jxr, */*
Accept-Language: ko
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; Trident/7.0; rv:11.0) like Gecko
Accept-Encoding: gzip, deflate
Host: 203.252.33.54:10080
Connection: Keep-Alive

HTTP/1.1 200 OK
```

- HTTP request, response message 를 매번 출력하도록 했다. 잘 출력되는 모습을 확인했다.
- 위가 받은 request message, 아래가 보낸 response message 이다.



- 직접 존재하는 file 을 URL 에 입력해 접근할 수 있음을 확인했다.



웹 페이지를 찾을 수 없습니다.

HTTP 404

예측 가능한 원인:

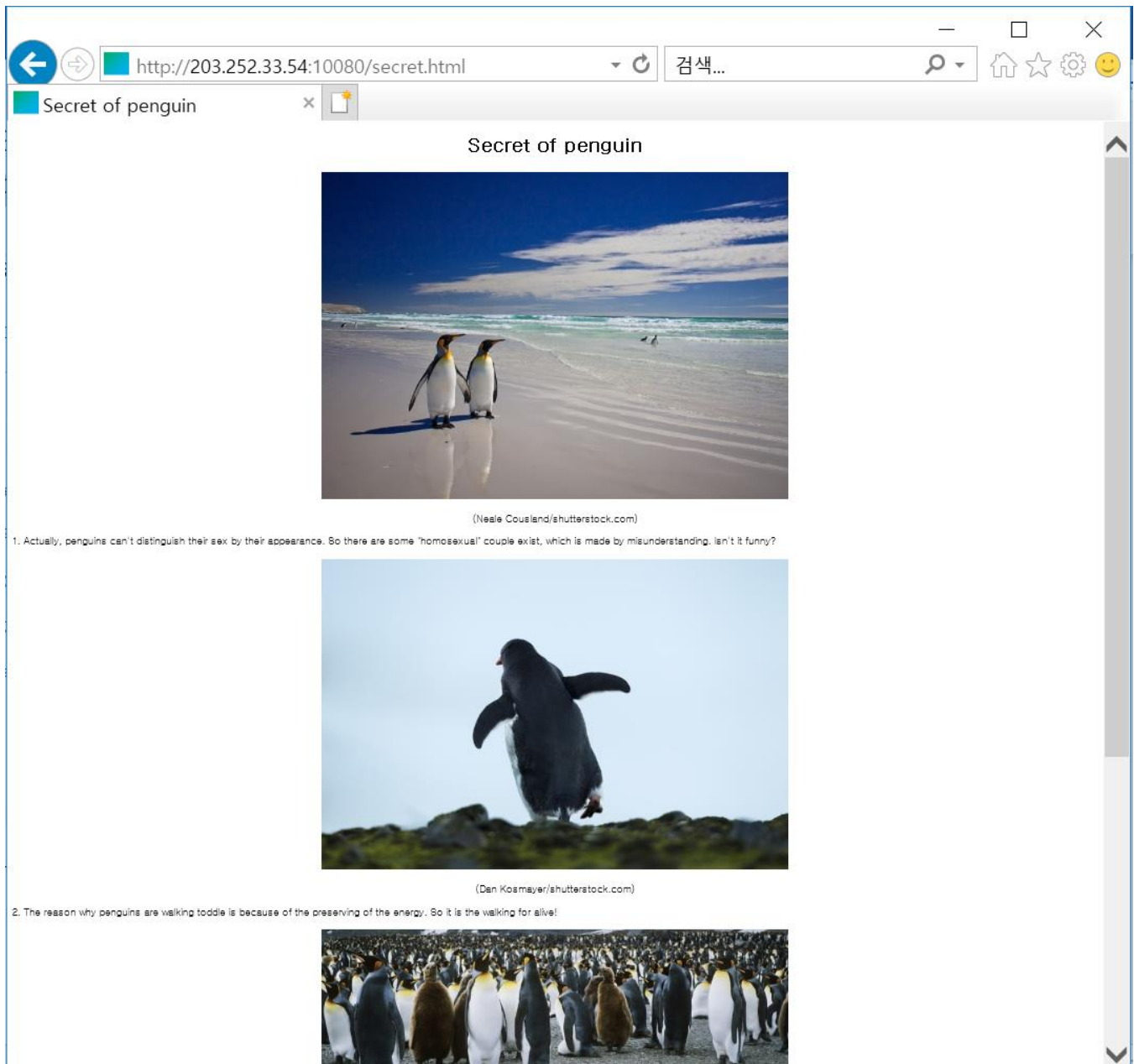
- 주소에 오타가 있을 수 있습니다.
- 클릭한 링크가 만료된 것일 수도 있습니다.

가능한 해결 방법:

- 주소를 다시 입력하십시오.
- 이전 페이지로 돌아갑니다.
- 기본 사이트로 원하는 정보를 찾습니다.
- 추가 정보

- 존재하지 않는 파일을 접근한 경우 HTTP 404 error 을 주는 것을 확인할 수 있다.





- 로그인 버튼을 통해 secret.html 파일에 들어간 경우

the short legs, their legs are quite long. Their total leg length is almost the half of their body!



←

→

http://203.252.33.54:10080/secret5.pdf

↻

검색...

🔍

🏠

☆

⚙️

🌙

203.252.33.54

✕

🖼️

# Introduction to Computer Networks

## Assignment 3: My Own Web Server

### 1. Goal

- Develop a simple web server using socket programming.

### 2. Development environments

- Secret.html 파일에서 동영상 재생 / 링크를 통한 pdf 파일 접근 등이 문제없이 잘 됨을 확인했다.

---

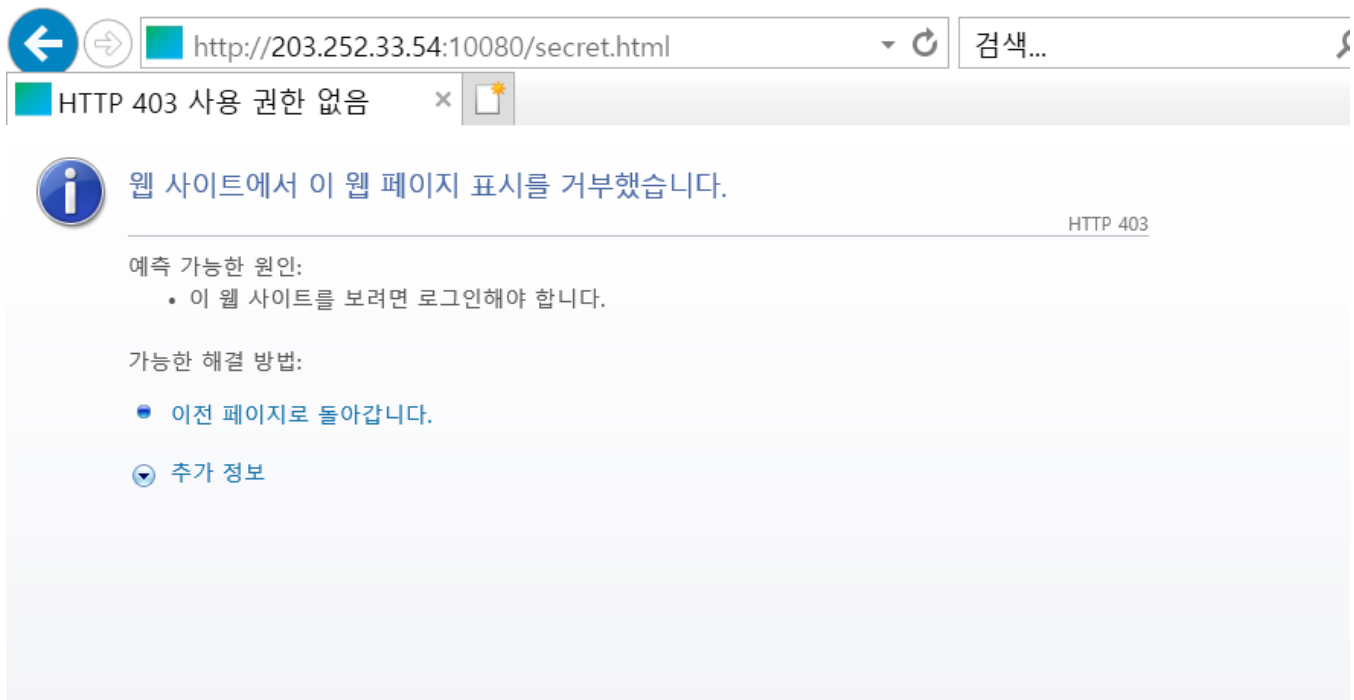
```
GET /secret.html HTTP/1.1
Accept: text/html, application/xhtml+xml, image/jxr, */*
Referer: http://203.252.33.54:10080/
Accept-Language: ko
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; Trident/7.0; rv:11.0) like Gecko
Accept-Encoding: gzip, deflate
Host: 203.252.33.54:10080
Connection: Keep-Alive
```

```
HTTP/1.1 200 OK
```

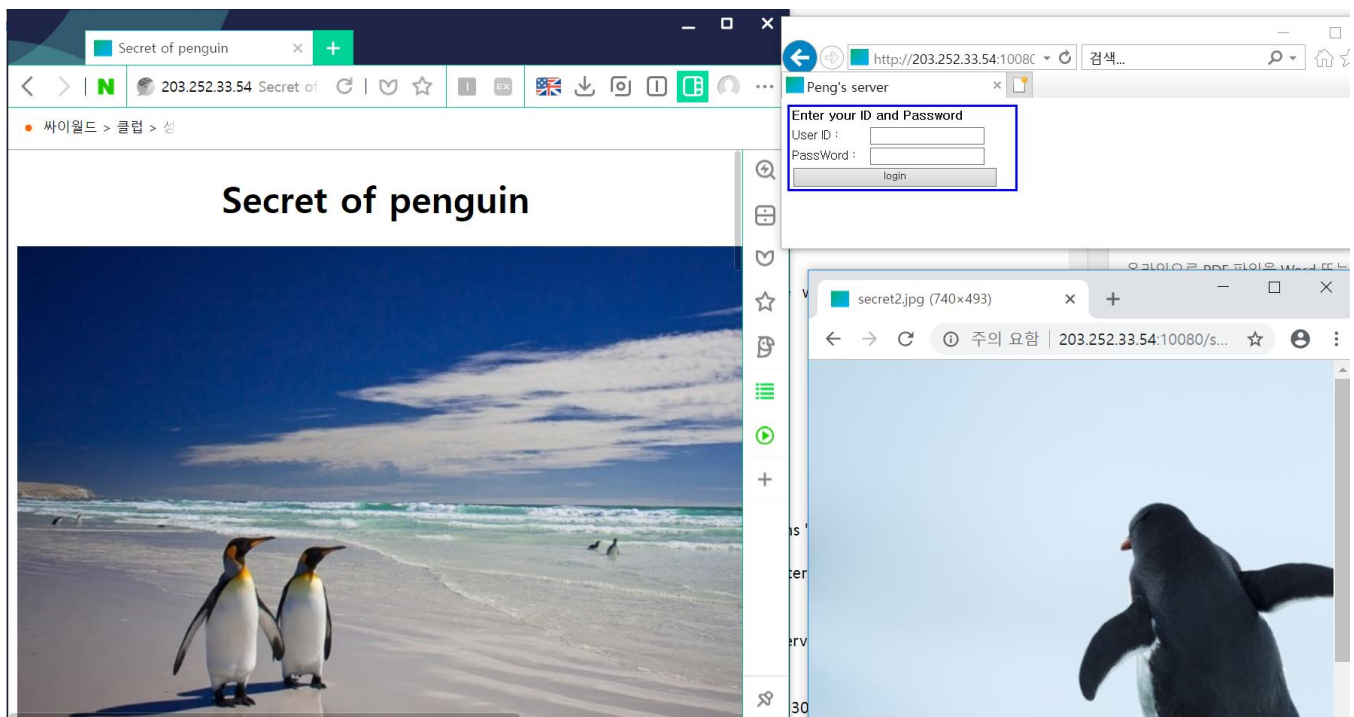
```
GET /secret1.jpg HTTP/1.1
Accept: image/png, image/svg+xml, image/jxr, image/*;q=0.8, */*;q=0.5
Referer: http://203.252.33.54:10080/secret.html
Accept-Language: ko
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; Trident/7.0; rv:11.0) like Gecko
Accept-Encoding: gzip, deflate
Host: 203.252.33.54:10080
Connection: Keep-Alive
```

```
HTTP/1.1 200 OK
```

- 또한 다음과 같이 secret.html 과 같은 html 내부에 파일이 있는 경우 해당 파일들을 자동으로 request 하는 것을 확인했다.



- secret.html 에 직접적으로 access 한 경우 HTTP 403 error 를 반환함을 확인했다.



- 다음과 같이 여러 종류의 browser 를 통해 접근해도 concurrent 하게 접속됨을 확인했다.