HTTPS

제출일	2022.11.30	**전공**	컴퓨터공학과
과목	컴퓨터네트워크(02)	**학번**	201602037
담당교수	이영석 교수님	**이름**	이규정

① Nginx 설정

Docker를 실행시킬 때 -v 옵션을 사용하여 로컬에서 만든 인증키를 container를 연결시켜주려고 했으나 포트번호 설정에서 자꾸 오류가 발생하여 rootCA.key와 talkCA.crt 를 docker 안에서 직접 생성하여 만들어주었습니다.

```
PS C:\Users\verac\PycharmProjects\Computer_network> docker exec -it mynginx bash
root@082d0c4cdf5a:/# cd etc/nginx
root@082d0c4cdf5a:/etc/nginx# ls
conf.d  fastcgi_params  mime.types  modules  nginx.conf  rootCA.key  scgi_params  talkCA.crt  uwsgi_params
```

Nginx.conf

```
    server {
        listen 8080 ssl;
        server_name talk.localhost.com;
#         ssl_certificate talkCA.crt;
#         ssl_certificate_key rootCA.key;
        ssl_certificate /etc/nginx/talkCA.crt;
        ssl_certificate_key /etc/nginx/rootCA.key;
        ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
        ssl_ciphers HIGH:!aNULL:!MD5;

        location /pastebin/api/ {
            proxy_pass http://host.docker.internal:8889/;
            proxy_set_header X-Real-IP $remote_addr;
            proxy_set_header Host $host;
            proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        }

        location /pastebin/ {
            proxy_pass http://host.docker.internal:8890/pastebin/;
            proxy_set_header X-Real-IP $remote_addr;
            proxy_set_header Host $host;
            proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        }

        error_page 497 https://$server_name$request_uri;
    }
#     server {
#         listen 443 ssl;
```

Server를 정의하는 부분을 하나 더 만들어서 해보려고 했으나, docker 구조와 서버간의 통신 구조에 익숙하지 않아서 기존의 8080포트에 ssl을 연결하는 식으로 해결하였습니다.

Ssl 인증서의 위치는 위에서 직접 만들어주었던 docker안의 경로로 설정해주었습니다.

1. App.py

```python
import json
import urllib
from flask import Flask, Blueprint, render_template, request, redirect
import ssl

ssl._create_default_https_context = ssl._create_unverified_context

#endpoint = 'https://localhost:8080/pastebin/api'
endpoint = 'https://talk.localhost.com:8080/pastebin/api'
app = Flask(__name__)
```

https 접속을 시도하면 CERTIFICATE_VERIFY_FAILED error가 발생하여 결과를 확인할 수 없었습니다. 검색하여 해결법을 찾아본 결과 위와 같이 코드 한 줄을 첨가하여 해결할 수 있었습니다.

Endpoint의 URL도 과제에 맞게 바꿔 주었습니다.

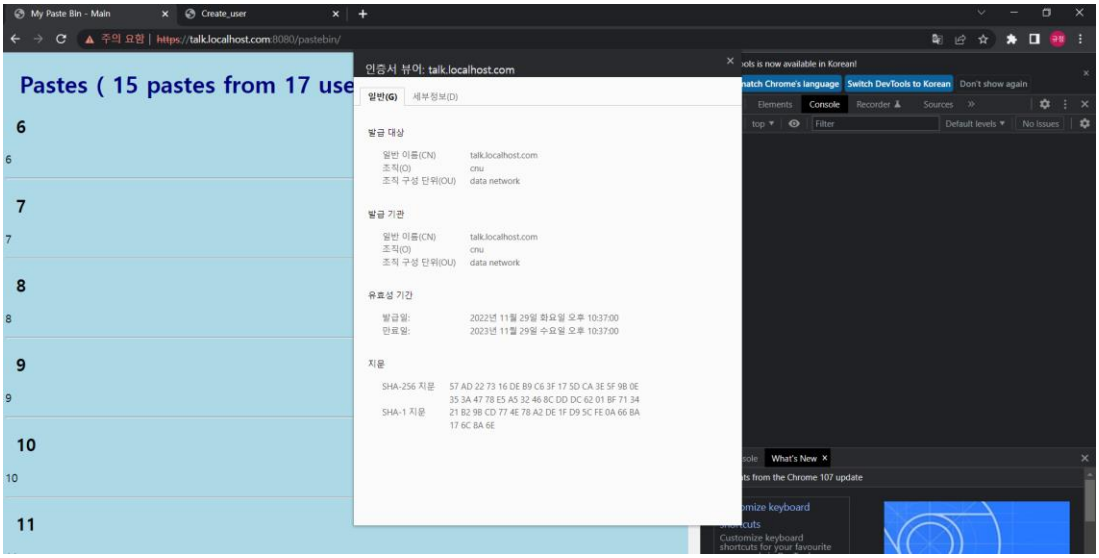
2. Myscript.js

```javascript
async function fetchRequestWithError() {
  try {
    skip_index = skip_index;
    //const url = 'https://localhost:8080/pastebin/api/pastes/?skip='+skip_index;
    const url = 'https://talk.localhost.com:8080/pastebin/api/pastes/?skip='+skip_index;
    const response = await fetch(url);
```

js 파일도 마찬가지로 URL 변경을 해주었습니다.

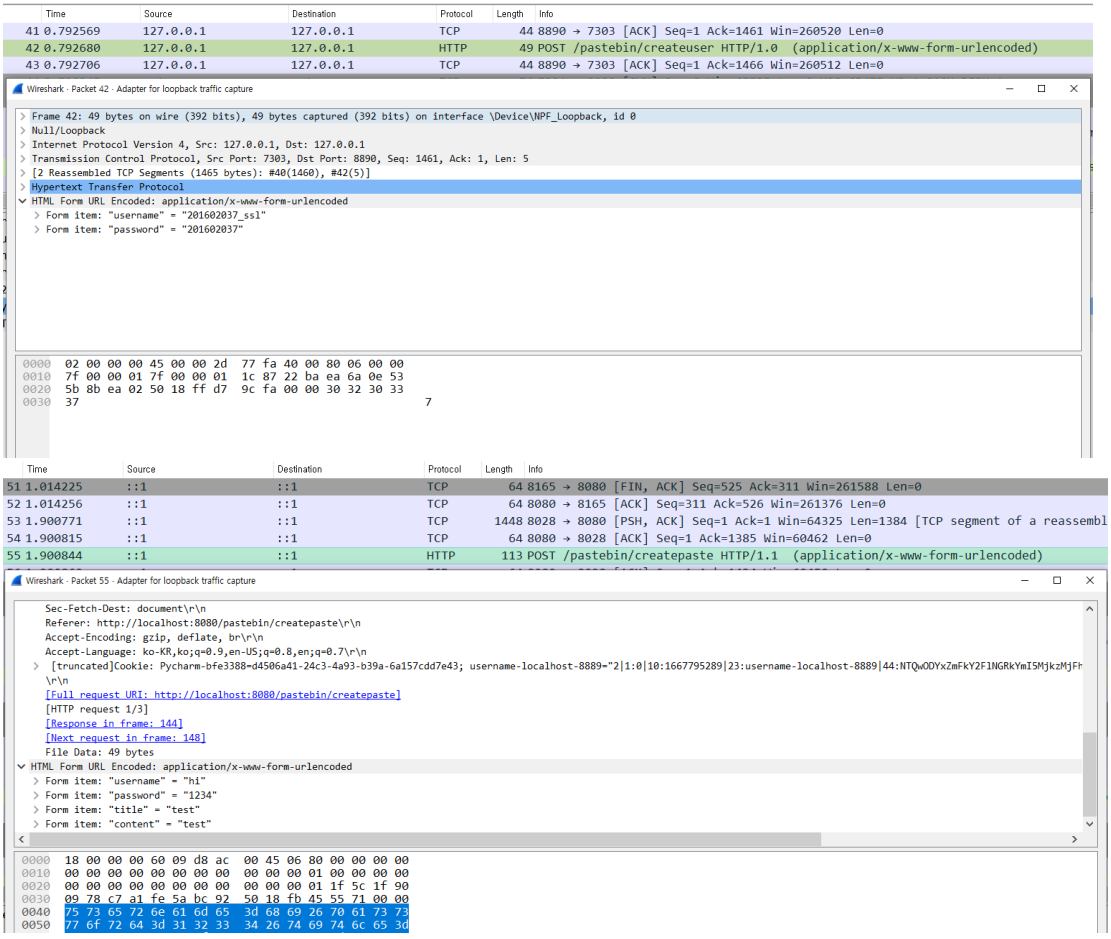
② 실행 결과

1. https://talk.localhost.com:8080/pastebin/



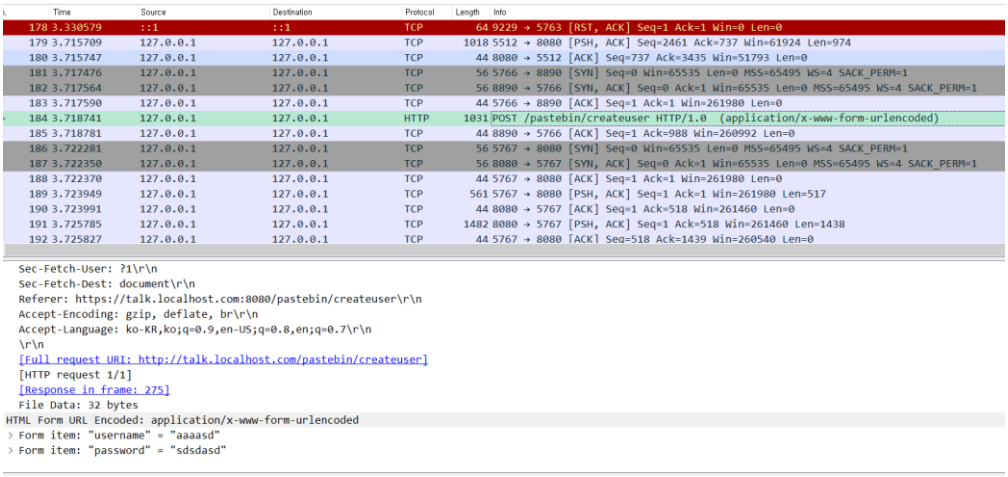
CORS 에러도 발생하지 않았고 https로 연결이 된 모습입니다.

2. 암호화 전 패킷 캡쳐



3. 암호화 후 패킷 캡쳐

https로 접속이 성공하고 패킷캡쳐를 해봤으나 암호화가 되지 않은 패킷이 캡쳐되었습니다. 이 부분은 해결하지 못했습니다.



	Time	Source	Destination	Protocol	Length	Info
178	3.330579	::1	::1	TCP	64	9229 → 5763 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
179	3.715709	127.0.0.1	127.0.0.1	TCP	1018	5512 → 8080 [PSH, ACK] Seq=2461 Ack=737 Win=61924 Len=974
180	3.715747	127.0.0.1	127.0.0.1	TCP	44	8080 → 5512 [ACK] Seq=737 Ack=3435 Win=51793 Len=0
181	3.717476	127.0.0.1	127.0.0.1	TCP	56	5766 → 8890 [SYN] Seq=0 Win=65535 Len=0 MSS=65495 WS=4 SACK_PERM=1
182	3.717564	127.0.0.1	127.0.0.1	TCP	56	8890 → 5766 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=65495 WS=4 SACK_PERM=1
183	3.717590	127.0.0.1	127.0.0.1	TCP	44	5766 → 8890 [ACK] Seq=1 Ack=1 Win=261980 Len=0
184	3.718741	127.0.0.1	127.0.0.1	HTTP	1031	POST /pastebin/createuser HTTP/1.0 (application/x-www-form-urlencoded)
185	3.718781	127.0.0.1	127.0.0.1	TCP	44	8890 → 5766 [ACK] Seq=1 Ack=988 Win=260992 Len=0
186	3.722281	127.0.0.1	127.0.0.1	TCP	56	5767 → 8080 [SYN] Seq=0 Win=65535 Len=0 MSS=65495 WS=4 SACK_PERM=1
187	3.722350	127.0.0.1	127.0.0.1	TCP	56	8080 → 5767 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=65495 WS=4 SACK_PERM=1
188	3.722370	127.0.0.1	127.0.0.1	TCP	44	5767 → 8080 [ACK] Seq=1 Ack=1 Win=261980 Len=0
189	3.723949	127.0.0.1	127.0.0.1	TCP	561	5767 → 8080 [PSH, ACK] Seq=1 Ack=1 Win=261980 Len=517
190	3.723991	127.0.0.1	127.0.0.1	TCP	44	8080 → 5767 [ACK] Seq=1 Ack=518 Win=261460 Len=0
191	3.725785	127.0.0.1	127.0.0.1	TCP	1482	8080 → 5767 [PSH, ACK] Seq=1 Ack=518 Win=261460 Len=1438
192	3.725827	127.0.0.1	127.0.0.1	TCP	44	5767 → 8080 [ACK] Seq=518 Ack=1439 Win=260540 Len=0

```
Sec-Fetch-User: ?1\r\n
Sec-Fetch-Dest: document\r\n
Referer: https://talk.localhost.com:8080/pastebin/createuser\r\n
Accept-Encoding: gzip, deflate, br\r\n
Accept-Language: ko-KR,ko;q=0.9,en-US;q=0.8,en;q=0.7\r\n
\r\n
[Full request URI: http://talk.localhost.com/pastebin/createuser]
[HTTP request 1/1]
[Response in frame: 275]
File Data: 32 bytes
HTML Form URL Encoded: application/x-www-form-urlencoded
> Form item: "username" = "aaaasd"
> Form item: "password" = "sdsdasd"
```