MSCS-631: Python: Lab5: Web Proxy

Sandesh Pokharel

University of the Cumberlands

MSCS-631 Advanced Computer Networks

Dr. Charles Lively

Feb 09, 2025

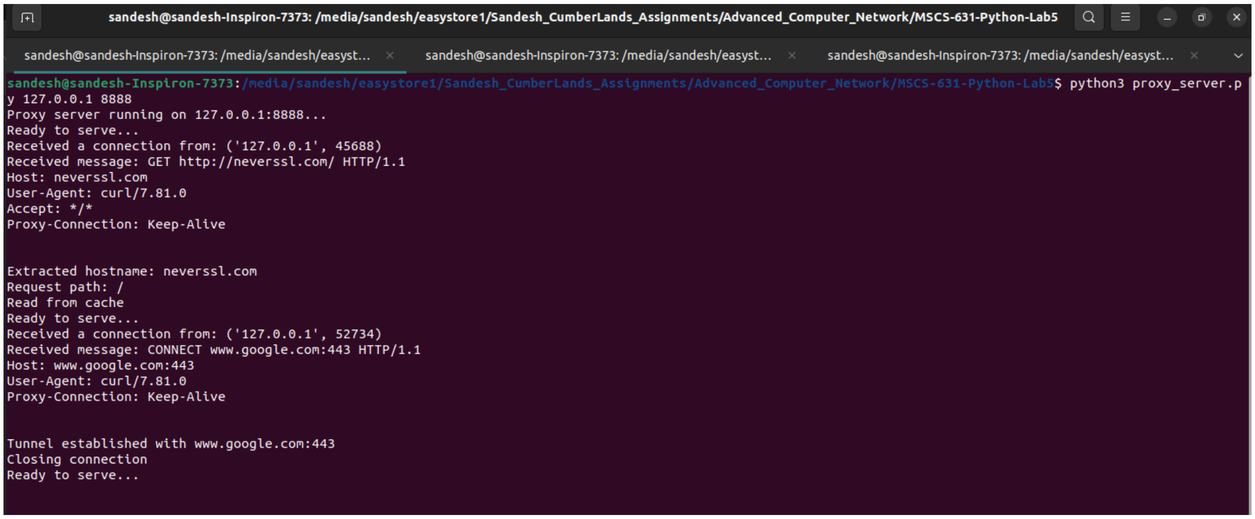
Challenges and Experiences

During this assignment, I gained valuable experience in setting up and testing a proxy server that handles both HTTP and HTTPS traffic. Initially, I struggled with configuring the server correctly to handle various requests due to errors such as "Empty reply from server" and issues with resolving certain hostnames. These challenges were compounded by the need to properly tunnel HTTPS requests, which involved correctly establishing secure TLS connections between the client and external servers. Once I resolved the hostname handling and configured the proxy to support the CONNECT method, I observed successful HTTPS responses from servers like Google, indicating that the tunnel had been properly established.

One of the key challenges was ensuring that the server responded with cached data for HTTP requests while also managing secure HTTPS tunnels. Debugging was necessary to understand why certain requests failed, particularly when the proxy encountered errors related to DNS lookups or refused connections. Another learning point was handling client certificates and ensuring the proxy could complete SSL handshakes. Despite these obstacles, I successfully demonstrated the proxy’s ability to fetch, cache, and serve both HTTP and HTTPS content, which was validated by successful curl requests and verified logs in the proxy server.

Screenshots

Screenshot 1: Running the proxy server

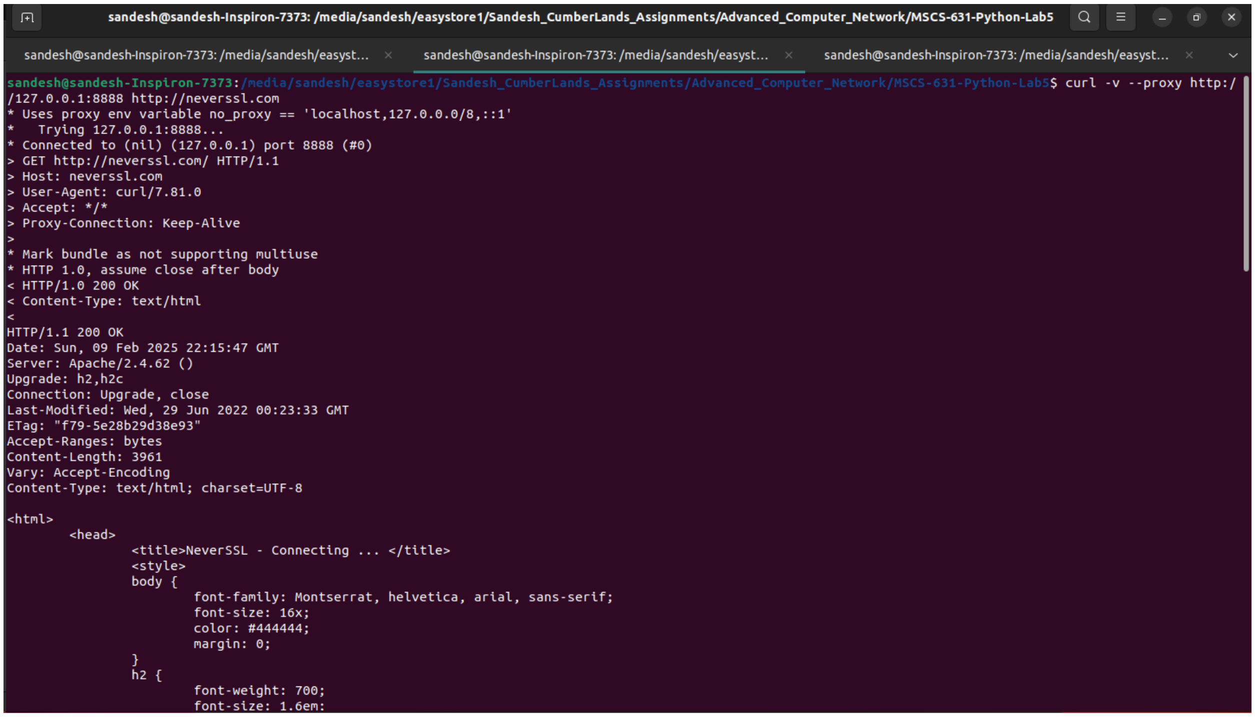

sandesh@sandesh-Inspiron-7373: /media/sandesh/easystore1/Sandesh_CumberLands_Assignments/Advanced_Computer_Network/MSCS-631-Python-Lab5
sandesh@sandesh-Inspiron-7373: /media/sandesh/easyst...
sandesh@sandesh-Inspiron-7373: /media/sandesh/easyst...
sandesh@sandesh-Inspiron-7373: /media/sandesh/easyst...

```
sandesh@sandesh-Inspiron-7373:/media/sandesh/easystore1/Sandesh_CumberLands_Assignments/Advanced_Computer_Network/MSCS-631-Python-Lab5$ python3 proxy_server.p
y 127.0.0.1 8888
Proxy server running on 127.0.0.1:8888...
Ready to serve...
Received a connection from: ('127.0.0.1', 45688)
Received message: GET http://neverssl.com/ HTTP/1.1
Host: neverssl.com
User-Agent: curl/7.81.0
Accept: */*
Proxy-Connection: Keep-Alive


Extracted hostname: neverssl.com
Request path: /
Read from cache
Ready to serve...
Received a connection from: ('127.0.0.1', 52734)
Received message: CONNECT www.google.com:443 HTTP/1.1
Host: www.google.com:443
User-Agent: curl/7.81.0
Proxy-Connection: Keep-Alive


Tunnel established with www.google.com:443
Closing connection
Ready to serve...
```

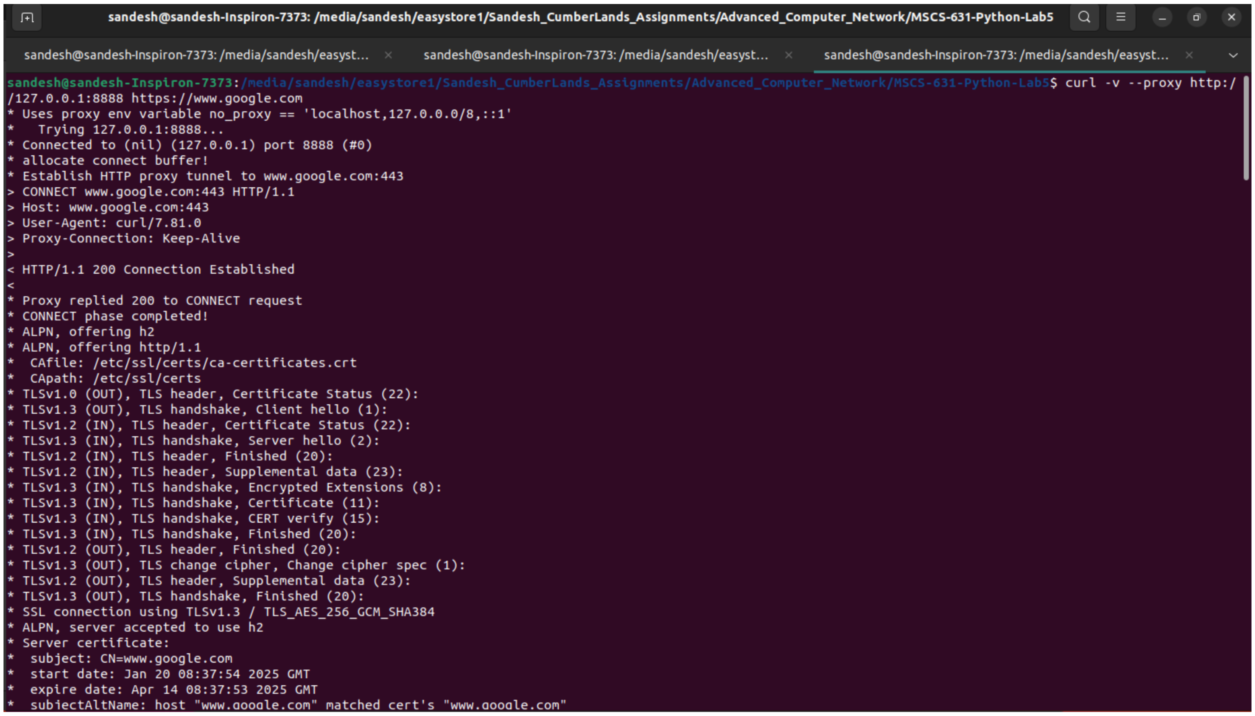
Screenshot 2: http request: neverssl.com


sandesh@sandesh-Inspiron-7373: /media/sandesh/easystore1/Sandesh_CumberLands_Assignments/Advanced_Computer_Network/MSCS-631-Python-Lab5
sandesh@sandesh-Inspiron-7373: /media/sandesh/easyst...
sandesh@sandesh-Inspiron-7373: /media/sandesh/easyst...
sandesh@sandesh-Inspiron-7373: /media/sandesh/easyst...

```
sandesh@sandesh-Inspiron-7373:/media/sandesh/easystore1/Sandesh_CumberLands_Assignments/Advanced_Computer_Network/MSCS-631-Python-Lab5$ curl -v --proxy http:/
/127.0.0.1:8888 http://neverssl.com
* Uses proxy env variable no_proxy == 'localhost,127.0.0.0/8,::1'
*   Trying 127.0.0.1:8888...
* Connected to (nil) (127.0.0.1) port 8888 (#0)
> GET http://neverssl.com/ HTTP/1.1
> Host: neverssl.com
> User-Agent: curl/7.81.0
> Accept: */*
> Proxy-Connection: Keep-Alive
>
* Mark bundle as not supporting multiuse
* HTTP 1.0, assume close after body
< HTTP/1.0 200 OK
< Content-Type: text/html
<
HTTP/1.1 200 OK
Date: Sun, 09 Feb 2025 22:15:47 GMT
Server: Apache/2.4.62 ()
Upgrade: h2,h2c
Connection: Upgrade, close
Last-Modified: Wed, 29 Jun 2022 00:23:33 GMT
ETag: "f79-5e28b29d38e93"
Accept-Ranges: bytes
Content-Length: 3961
Vary: Accept-Encoding
Content-Type: text/html; charset=UTF-8

<html>
        <head>
                <title>NeverSSL - Connecting ... </title>
                <style>
                body {
                        font-family: Montserrat, helvetica, arial, sans-serif;
                        font-size: 16x;
                        color: #444444;
                        margin: 0;
                }
                h2 {
                        font-weight: 700;
                        font-size: 1.6em;
```

Screentshot 3: https request: google.com


sandesh@sandesh-Inspiron-7373: /media/sandesh/easystore1/Sandesh_CumberLands_Assignments/Advanced_Computer_Network/MSCS-631-Python-Lab5
sandesh@sandesh-Inspiron-7373: /media/sandesh/easyst...
sandesh@sandesh-Inspiron-7373: /media/sandesh/easyst...
sandesh@sandesh-Inspiron-7373: /media/sandesh/easyst...

```
sandesh@sandesh-Inspiron-7373:/media/sandesh/easystore1/Sandesh_CumberLands_Assignments/Advanced_Computer_Network/MSCS-631-Python-Lab5$ curl -v --proxy http:/
/127.0.0.1:8888 https://www.google.com
* Uses proxy env variable no_proxy == 'localhost,127.0.0.0/8,::1'
*   Trying 127.0.0.1:8888...
* Connected to (nil) (127.0.0.1) port 8888 (#0)
* allocate connect buffer!
* Establish HTTP proxy tunnel to www.google.com:443
> CONNECT www.google.com:443 HTTP/1.1
> Host: www.google.com:443
> User-Agent: curl/7.81.0
> Proxy-Connection: Keep-Alive
>
< HTTP/1.1 200 Connection Established
<
* Proxy replied 200 to CONNECT request
* CONNECT phase completed!
* ALPN, offering h2
* ALPN, offering http/1.1
*  CAfile: /etc/ssl/certs/ca-certificates.crt
*  CApath: /etc/ssl/certs
* TLSv1.0 (OUT), TLS header, Certificate Status (22):
* TLSv1.3 (OUT), TLS handshake, Client hello (1):
* TLSv1.2 (IN), TLS header, Certificate Status (22):
* TLSv1.3 (IN), TLS handshake, Server hello (2):
* TLSv1.2 (IN), TLS header, Finished (20):
* TLSv1.2 (IN), TLS header, Supplemental data (23):
* TLSv1.3 (IN), TLS handshake, Encrypted Extensions (8):
* TLSv1.3 (IN), TLS handshake, Certificate (11):
* TLSv1.3 (IN), TLS handshake, CERT verify (15):
* TLSv1.3 (IN), TLS handshake, Finished (20):
* TLSv1.2 (OUT), TLS header, Finished (20):
* TLSv1.3 (OUT), TLS change cipher, Change cipher spec (1):
* TLSv1.2 (OUT), TLS header, Supplemental data (23):
* TLSv1.3 (OUT), TLS handshake, Finished (20):
* SSL connection using TLSv1.3 / TLS_AES_256_GCM_SHA384
* ALPN, server accepted to use h2
* Server certificate:
*  subject: CN=www.google.com
*  start date: Jan 20 08:37:54 2025 GMT
*  expire date: Apr 14 08:37:53 2025 GMT
*  subjectAltName: host "www.google.com" matched cert's "www.google.com"
```