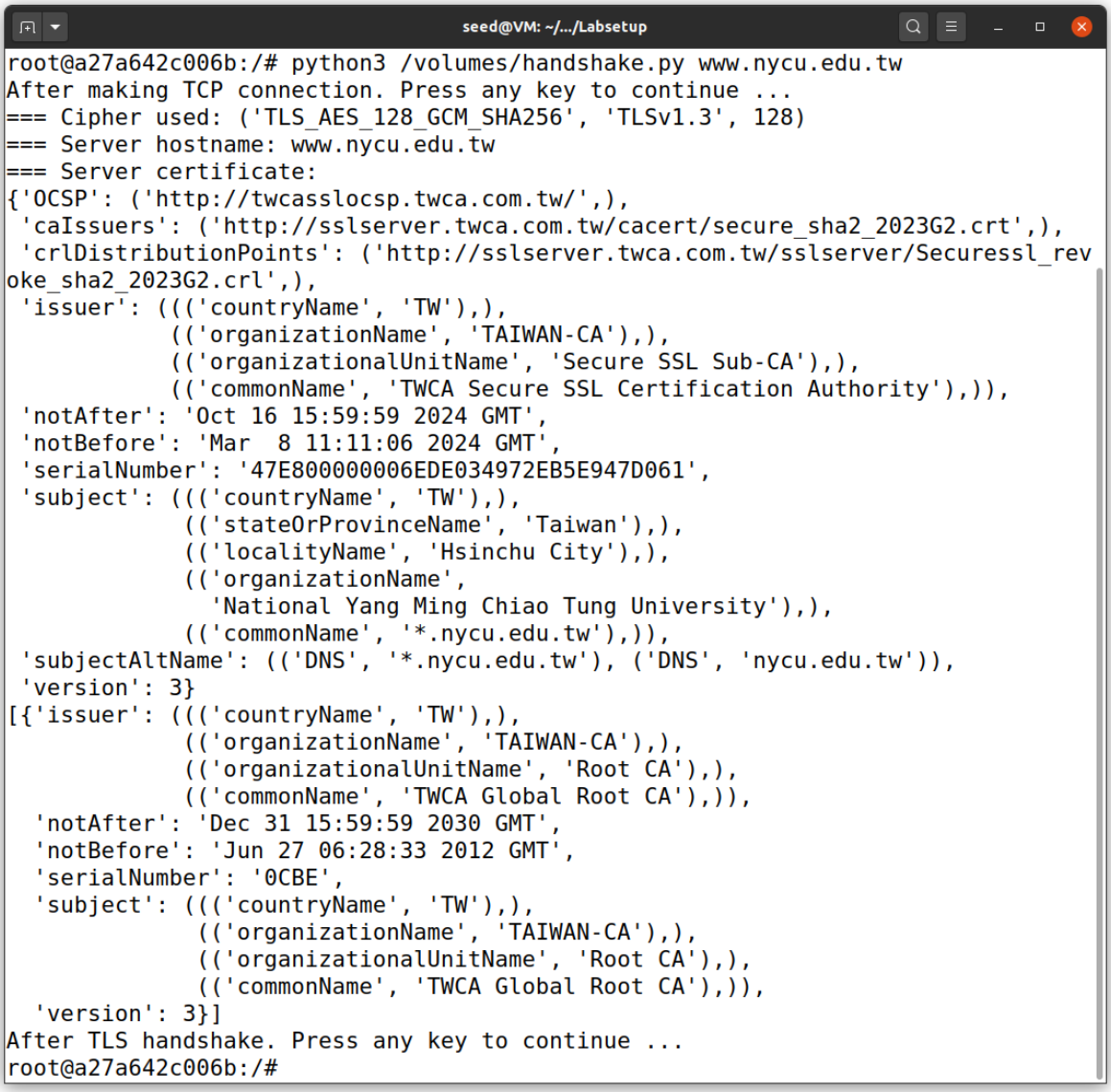
姓名：蔡佩蓉　　學號：109511286

Transport Layer Security (TLS) Lab (Lab6)

Task 1: TLS Client

Task 1.1: TLS handshake

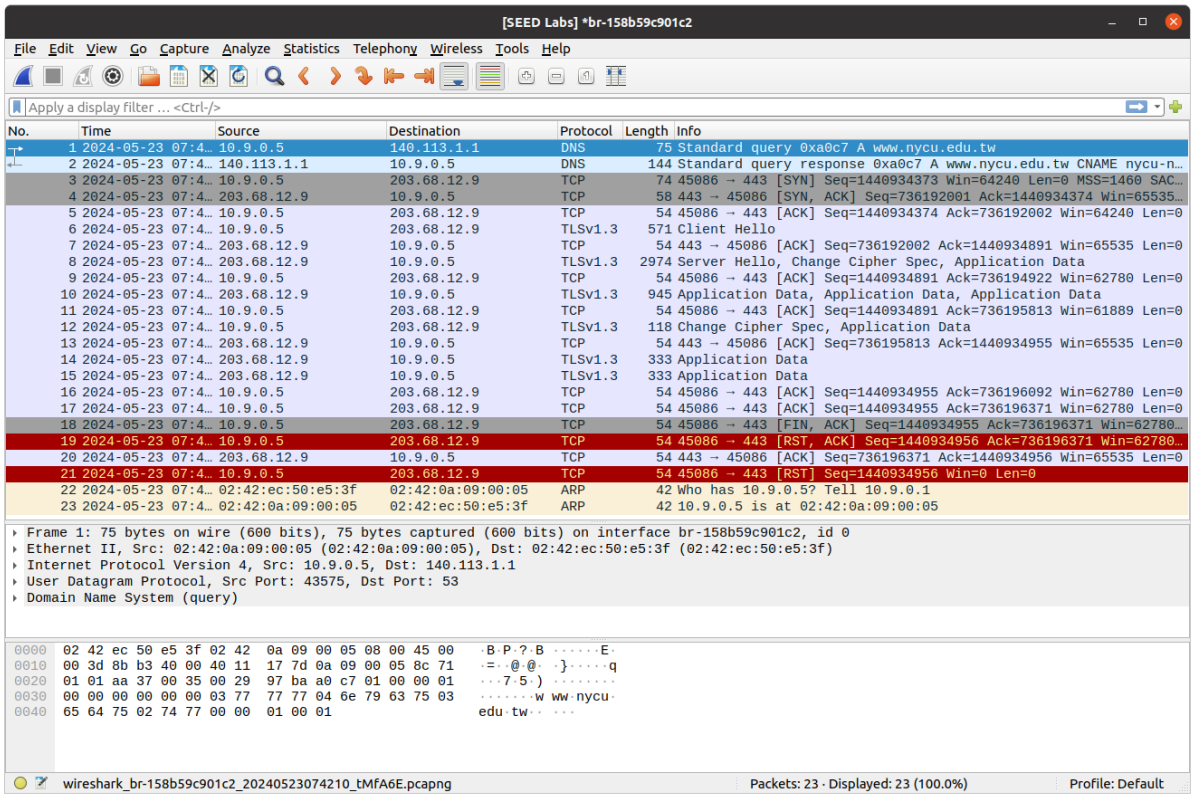


1. **What is the cipher used between the client and the server?**
   Cipher used: ('TLS_AES_128_GCM_SHA256', 'TLSv1.3', 128)

2. **Please print out the server certificate in the program.**
   Shown above.

3. **Explain the purpose of /etc/ssl/certs.**
   The directory /etc/ssl/certs contains a collection of root (trusted) CA (Certificate Authority) certificates. These certificates are used by the client to verify the authenticity of the server's certificate. By specifying this directory, the script ensures that it only trusts certificates issued by recognized CAs stored in this location.

4. **Use Wireshark to capture the network traffics during the execution of the program, and explain your observation. In particular, explain which step triggers the TCP handshake, and which step triggers the TLS handshake. Explain the relationship between the TLS handshake and the TCP handshake.**

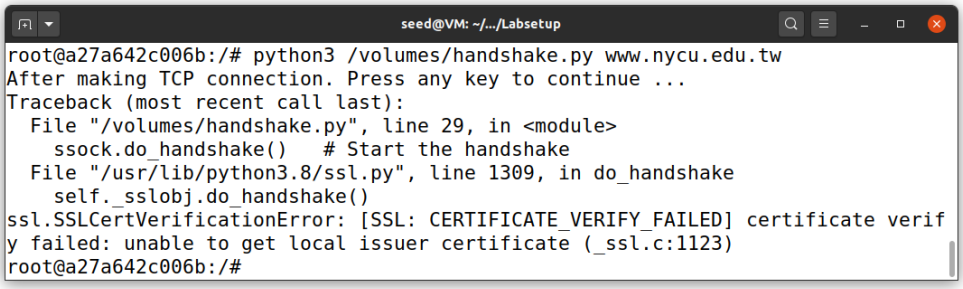


The TCP handshake consists of three packets: SYN, SYN-ACK, and ACK [index 3 to 5], while the TLS handshake involves multiple packets. [index 6 to 13]

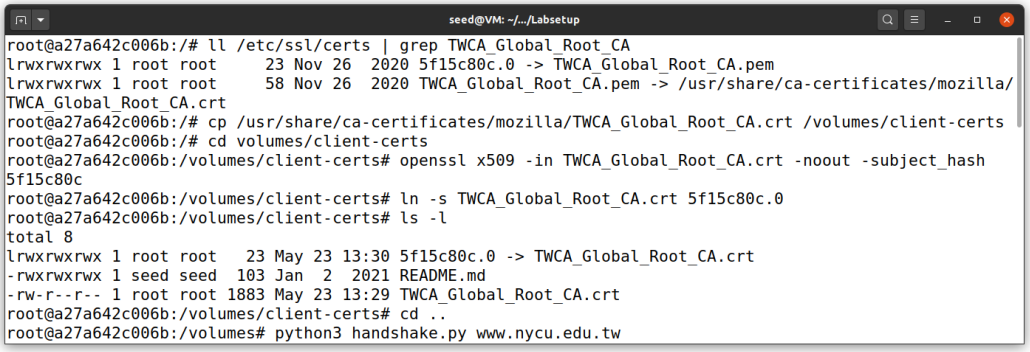
The client first sends out Client Hello related data, including all cipher suites supported by the client, client random numbers (used as Nonce), and etc. The server then replies to the client Server Hello related data and the certificate. Subsequently, the client verifies the validity of the certificate. If it is valid, the client sends a Key Exchange Client Key Exchange and a Change Cipher Spec (change cipher specification message) to the server, and the client handshake ends. The server replies to Change Cipher Spec (change cipher specification message) and a New Session value, and the handshake ends.

The TLS handshake occurs after the TCP handshake. The TCP handshake establishes a basic, reliable connection, and the TLS handshake builds on this connection to negotiate security parameters and establish an encrypted session. Essentially, the TLS handshake requires the underlying TCP connection to be established first, ensuring that the communication is both reliable and secure.

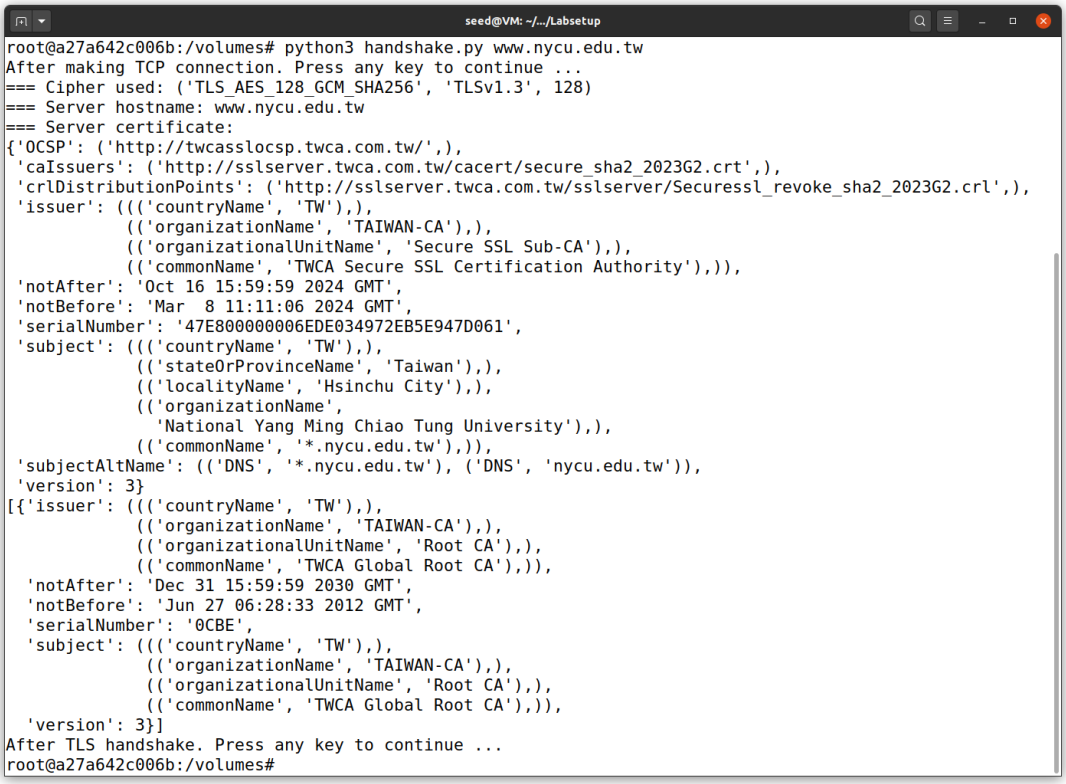
Task 1.2: CA's Certificate



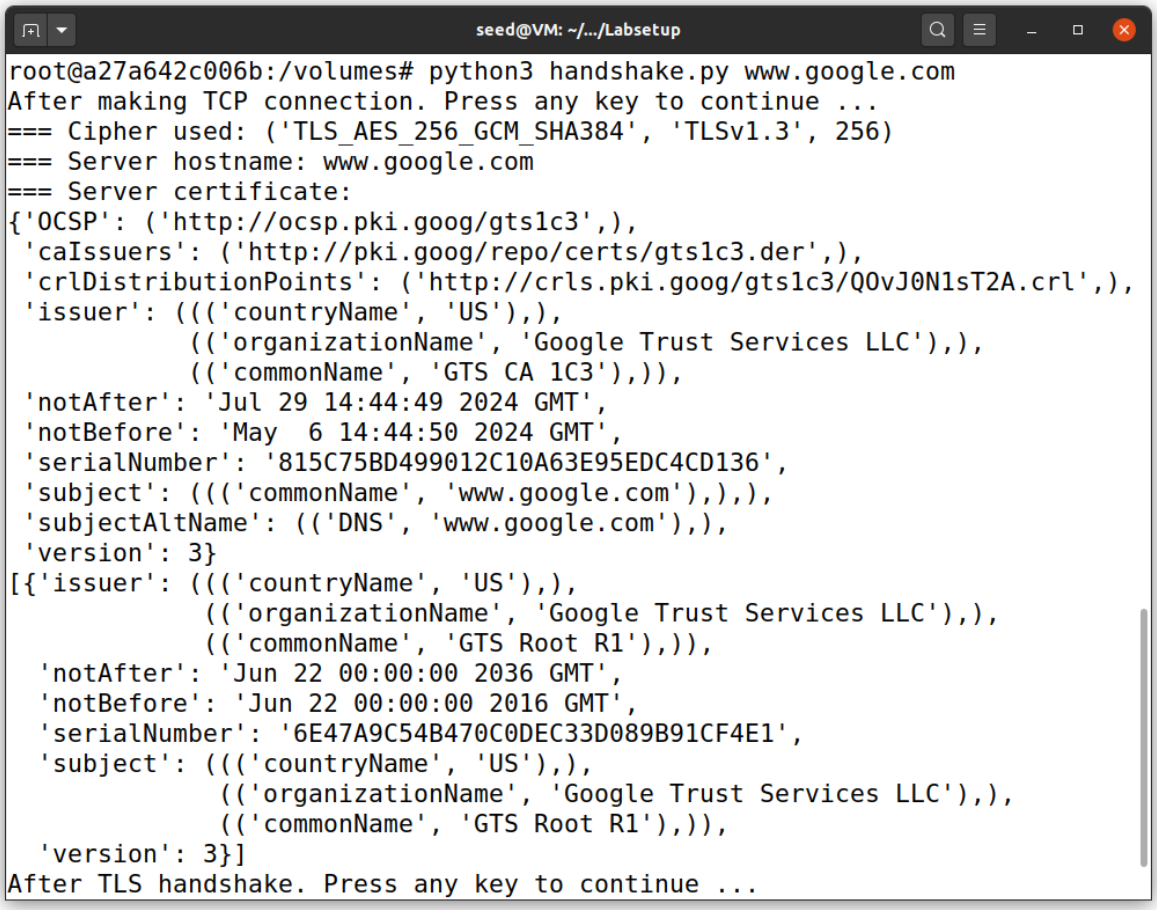
As observed, TLS failed. This is because the corresponding CA certificate was not found in ./client-certs.

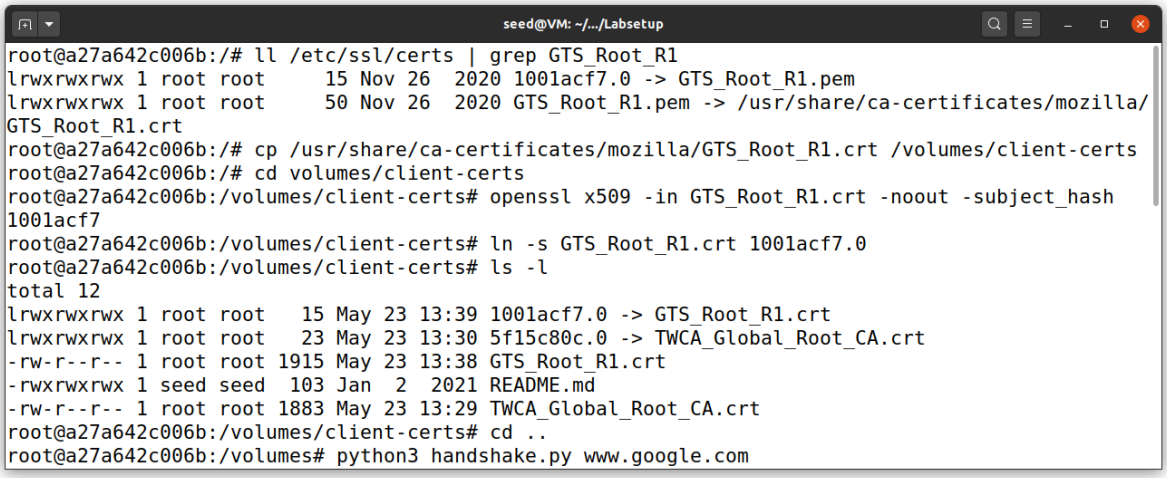


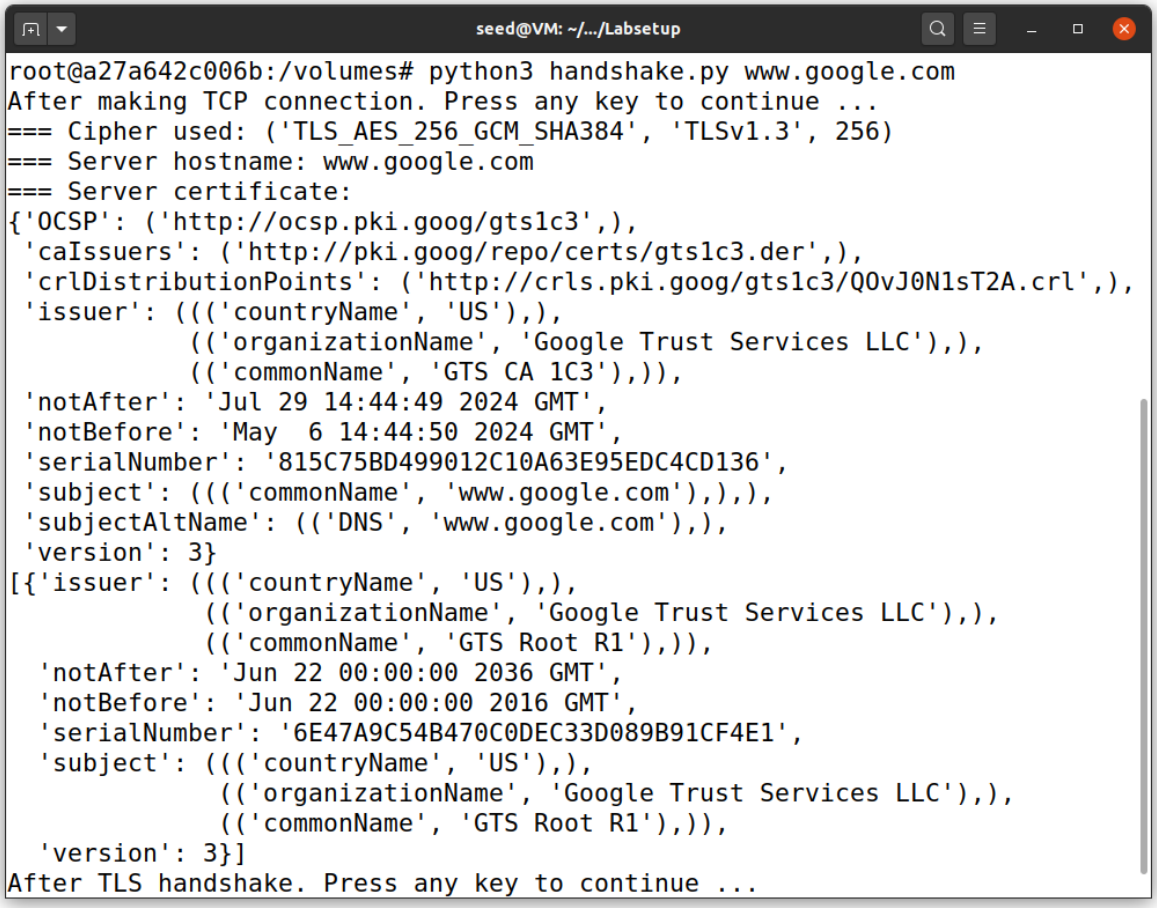
Identify the required CA certificate to verify the server (www.nycu.edu.tw) certificate, and then copy the certificate from the /etc/ssl/certs into ./client-certs. Simply copying the CA certificate into the client-certs directory does not create the necessary hashed symbolic links that the TLS client expects. Without these links, the client cannot locate the CA certificate needed to verify the server's certificate.



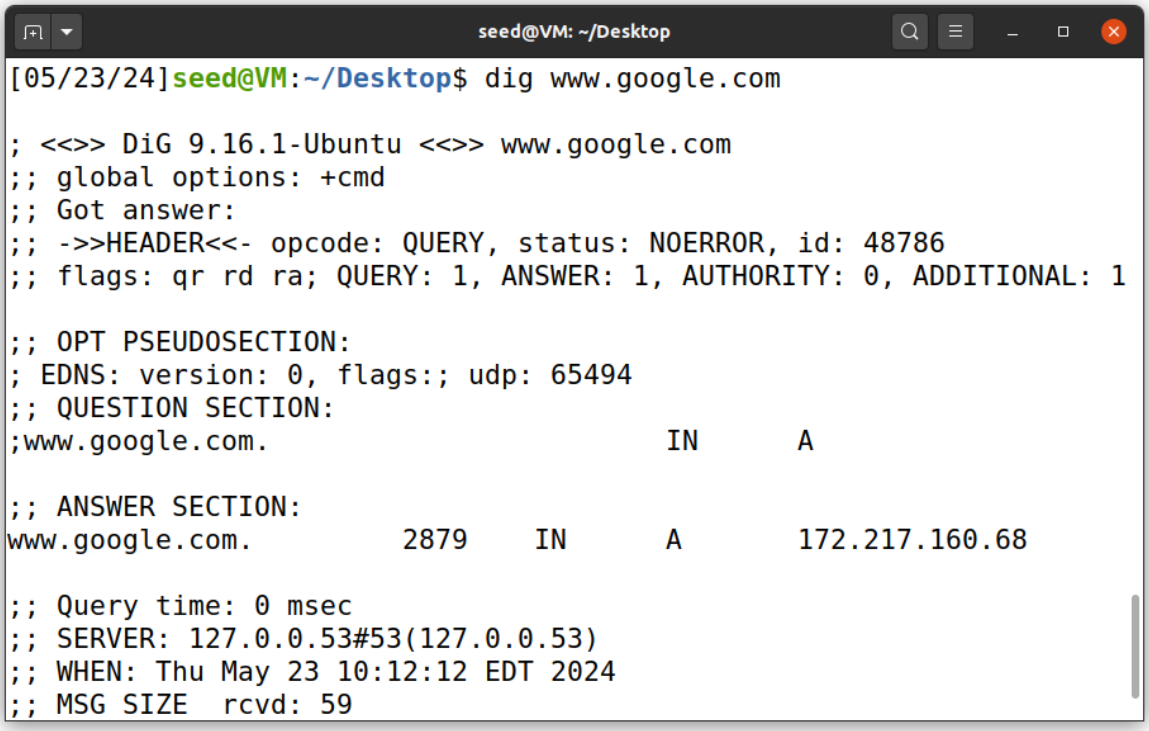
Additional requirement:



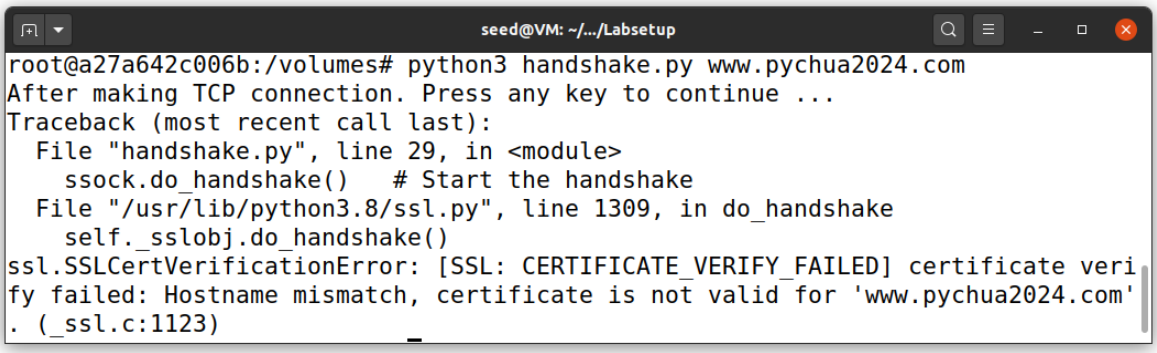




Task 1.3: Experiment with the hostname check

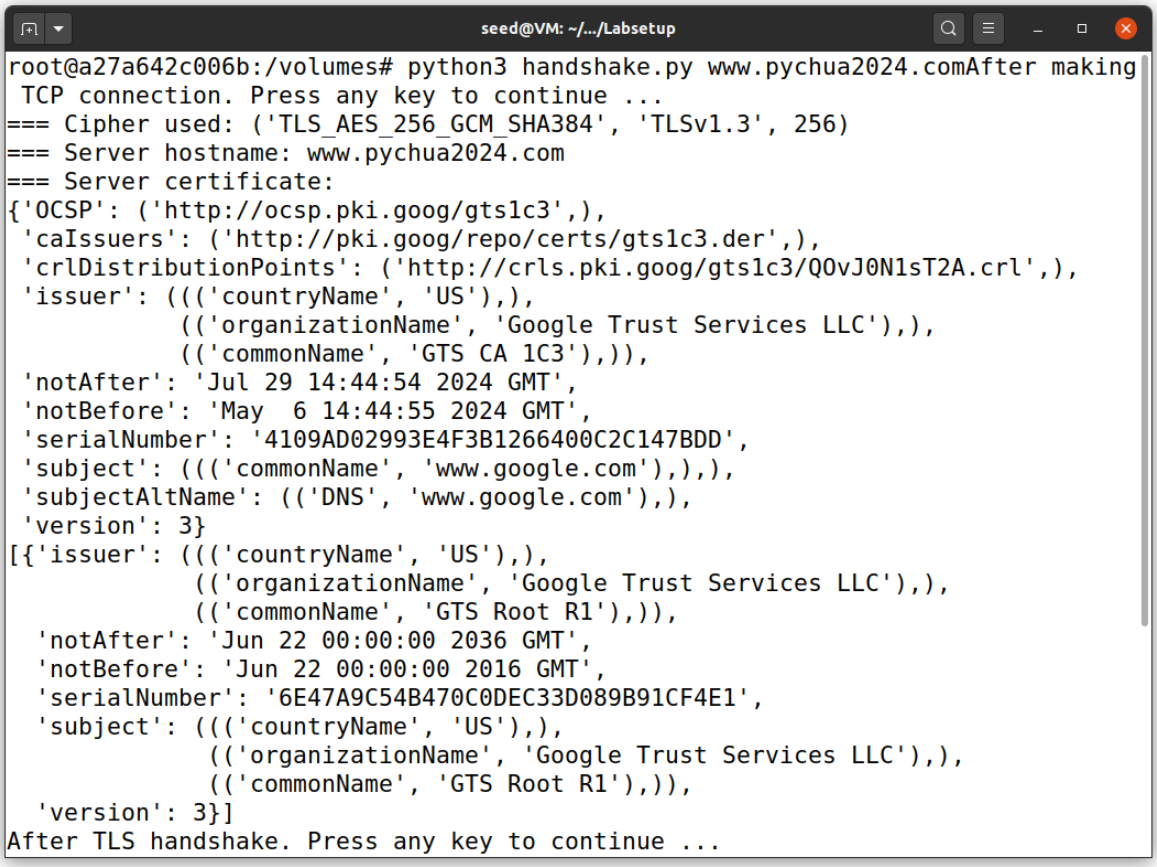


When context.check_hostname = True:



Observation: The client program will fail to connect to www.pychua2024.com, raising an ssl.SSLCertVerificationError indicating a hostname mismatch.
Explanation: Enabling hostname verification ensures that the server certificate matches the hostname the client is trying to connect to (www. pychua2024.com). Since the certificate is for www.google.com, the verification fails, and the client does not establish a connection.

When context.check_hostname = False:



Observation: The client program connects to www.pychua2024.com without verifying if the hostname in the server certificate matches www. pychua2024.com. This connection will succeed as long as the server certificate is valid (even if the hostname doesn't match).
Explanation: Disabling hostname verification means the client doesn't check if the certificate's Common Name (CN) or Subject Alternative Name (SAN) fields match the hostname it connects to. The connection succeeds if the certificate is otherwise valid.

Hostname verification is a critical security measure in TLS/SSL for the following reasons:

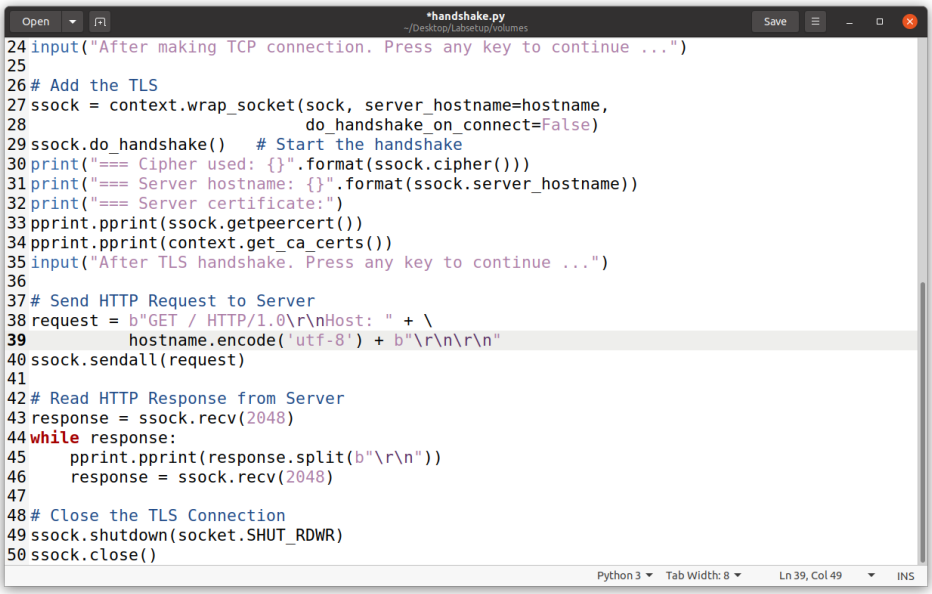
- Preventing Man-in-the-Middle (MitM) Attacks: Without hostname verification, an attacker could intercept traffic by presenting a valid certificate for a different domain. This would allow them to decrypt, read, and modify the data being transmitted.
- Ensuring Authenticity: Hostname checks ensure that the client is actually connecting to the intended server and not an imposter. This prevents spoofing attacks where a malicious server could pretend to be a legitimate server.
- Trustworthiness of Communication: Verifying the hostname helps to ensure the integrity and trustworthiness of the communication channel, making sure that the data is transmitted to and received from the correct source.

If the client program does not perform hostname verification, it is vulnerable to various security threats:

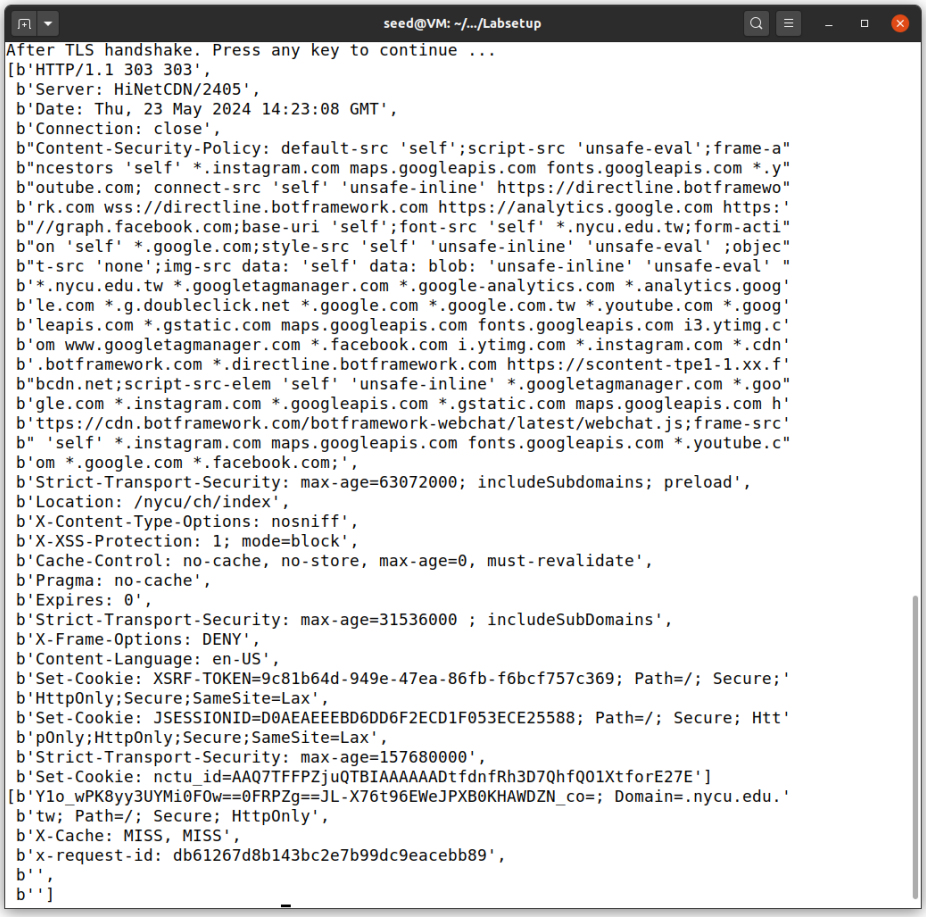
- Impersonation: Attackers could present valid certificates from other domains to impersonate the intended server, leading to unauthorized data access.
- Data Interception: Attackers could perform MitM attacks, intercepting and potentially altering the communication between the client and the legitimate server.
- Loss of Data Confidentiality and Integrity: Without hostname verification, the security guarantees of TLS/SSL are significantly weakened, leading to potential breaches of data confidentiality and integrity.

Task 1.4: Sending and getting Data

1. **Please add the data sending/receiving code to your client program, and report your observation.**

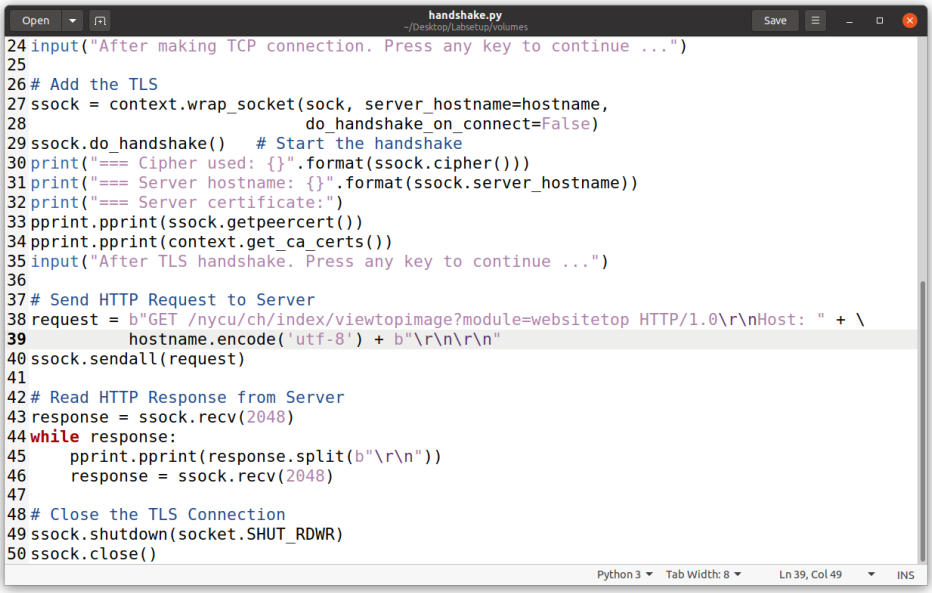


python3 handshake.py www.nycu.edu.tw

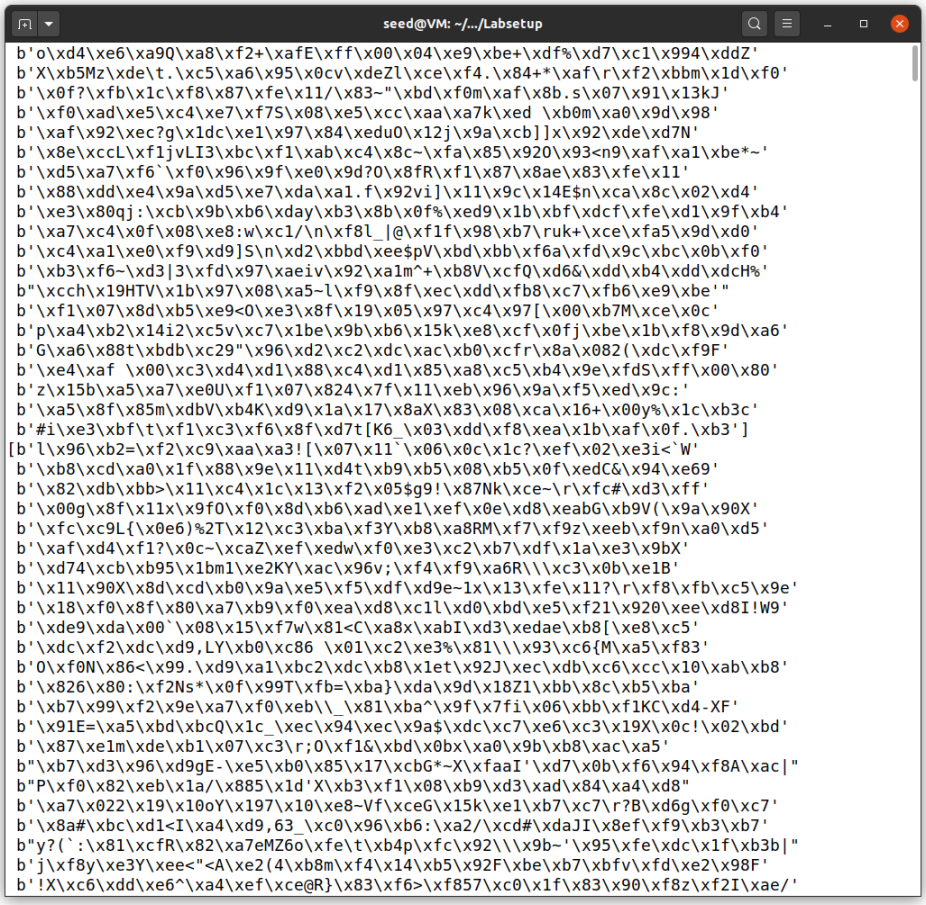


Obtained the source code (html) of www.nycu.edu.tw.

2. **Please modify the HTTP request, so you can fetch an image file of your choice from an HTTPS server (there is no need to display the image).**



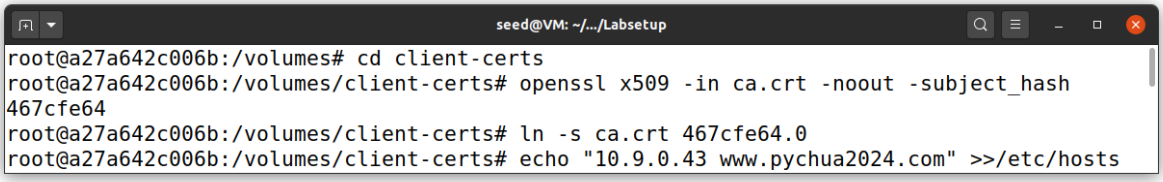
python3 handshake.py www.nycu.edu.tw (too long, thus only screenshot part of it)



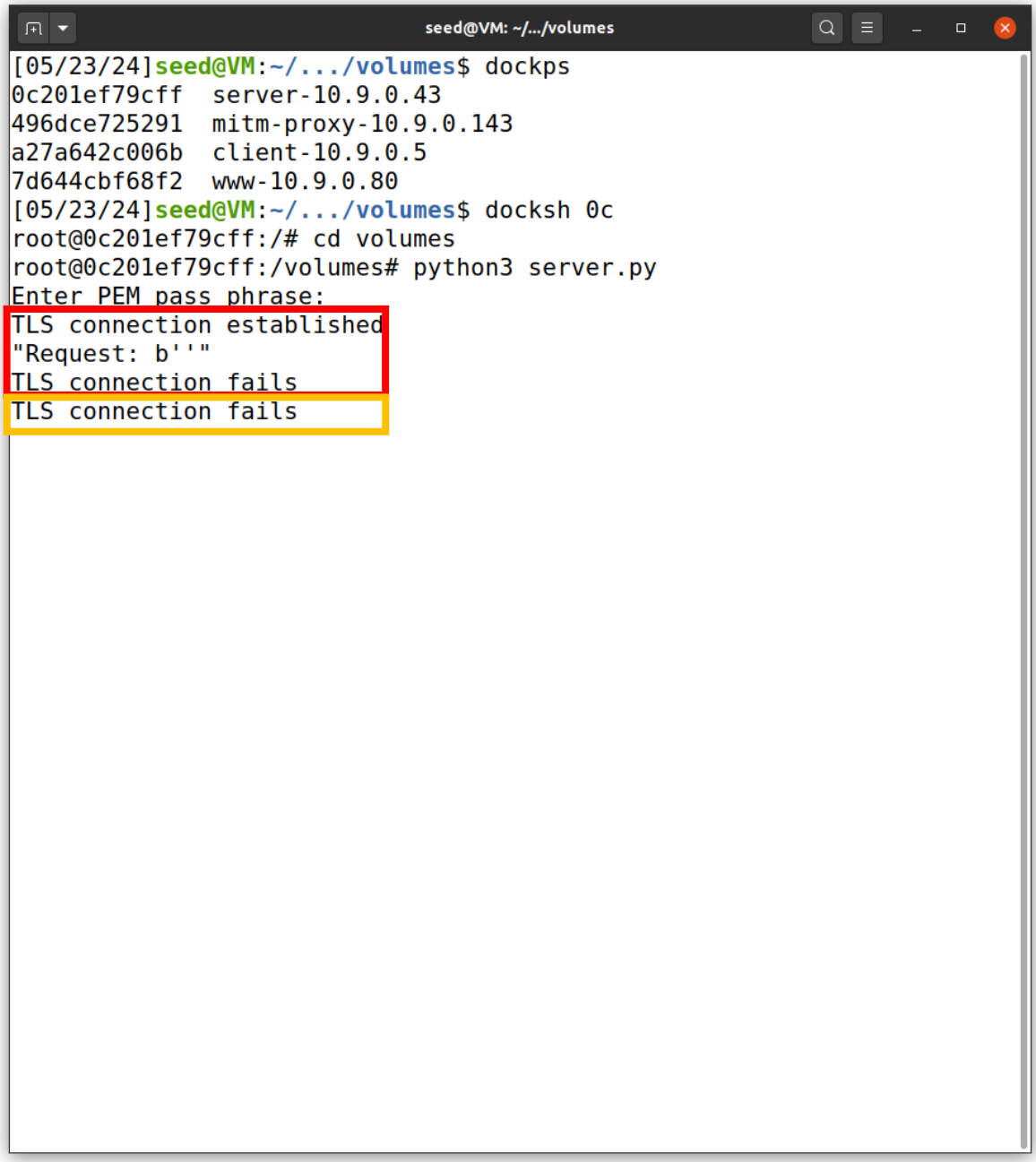
The server will respond with the image data, which will be printed as part of the HTTP response. However, since we're not displaying the image in the client program, the output will consist of the HTTP response headers followed by the binary image data. Observed that the binary image data will be printed as a list of byte strings.

Task 2: TLS Server

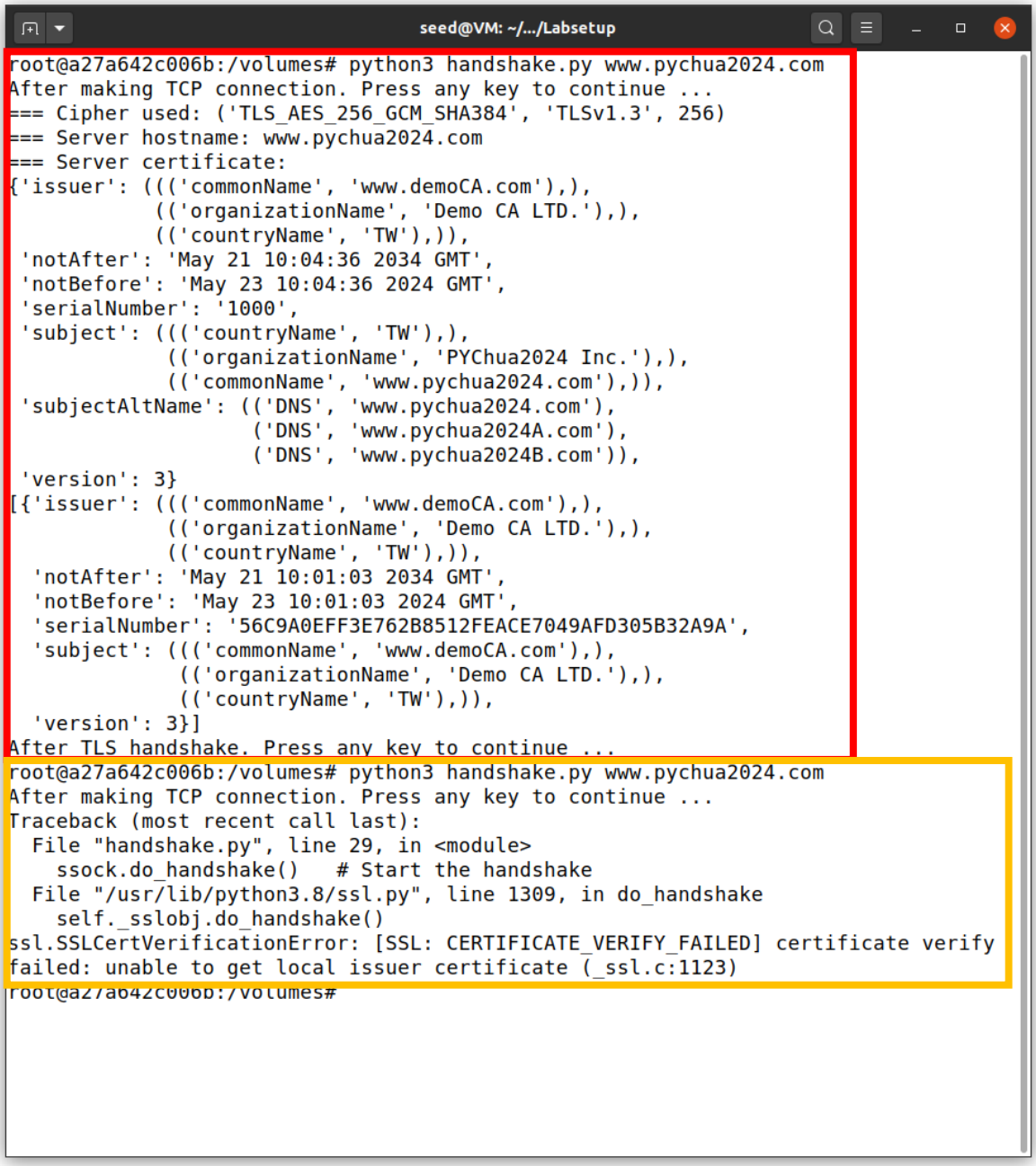
Task 2.1. Implement a simple TLS server



Server:

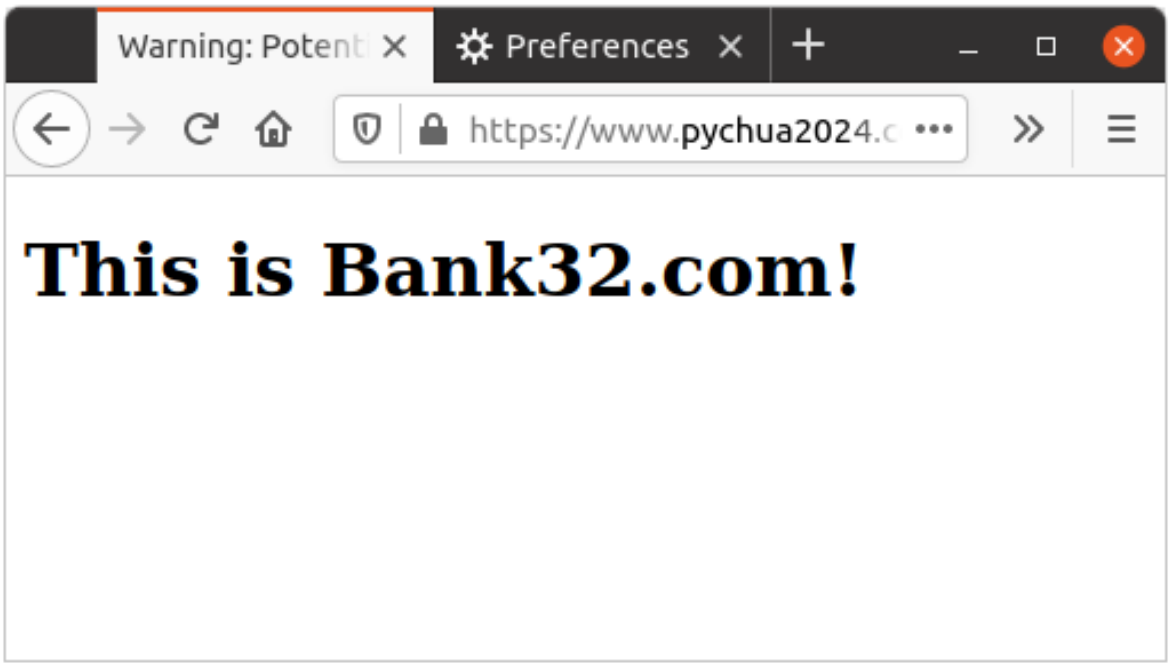

seed@VM: ~/.../volumes
[05/23/24]seed@VM:~/.../volumes$ dockps
0c201ef79cff server-10.9.0.43
496dce725291 mitm-proxy-10.9.0.143
a27a642c006b client-10.9.0.5
7d644cbf68f2 www-10.9.0.80
[05/23/24]seed@VM:~/.../volumes$ docksh 0c
root@0c201ef79cff:/# cd volumes
root@0c201ef79cff:/volumes# python3 server.py
Enter PEM pass phrase:
TLS connection established
"Request: b''"
TLS connection fails
TLS connection fails

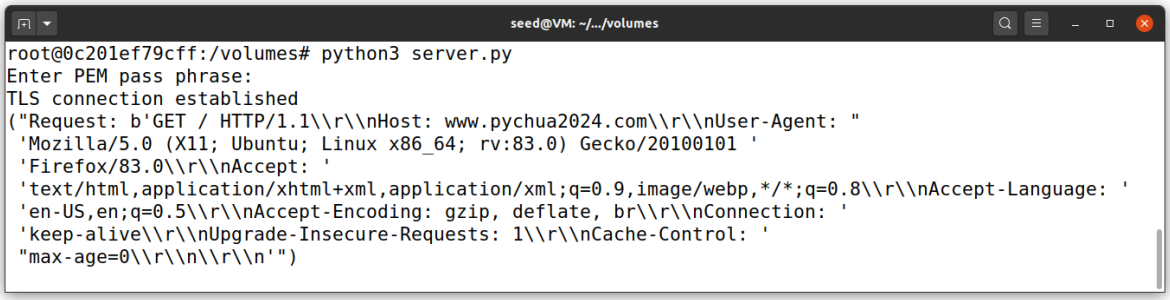
Client:



Red box indicates when cadir = './client-certs'; orange box indicates when cadir = '/etc/ssl/certs'.
Cannot find a certificate for www.pychua2024.com in cadir = '/etc/ssl/certs' thus no TLS connection.

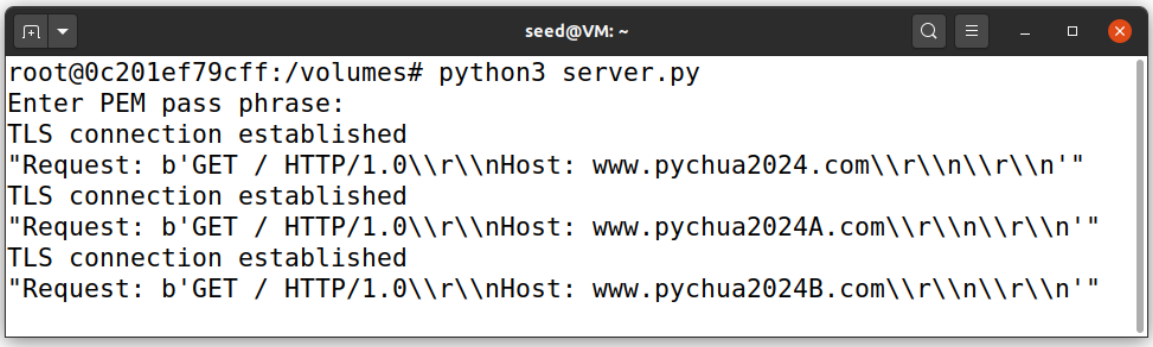
Task 2.2. Testing the server program using browsers



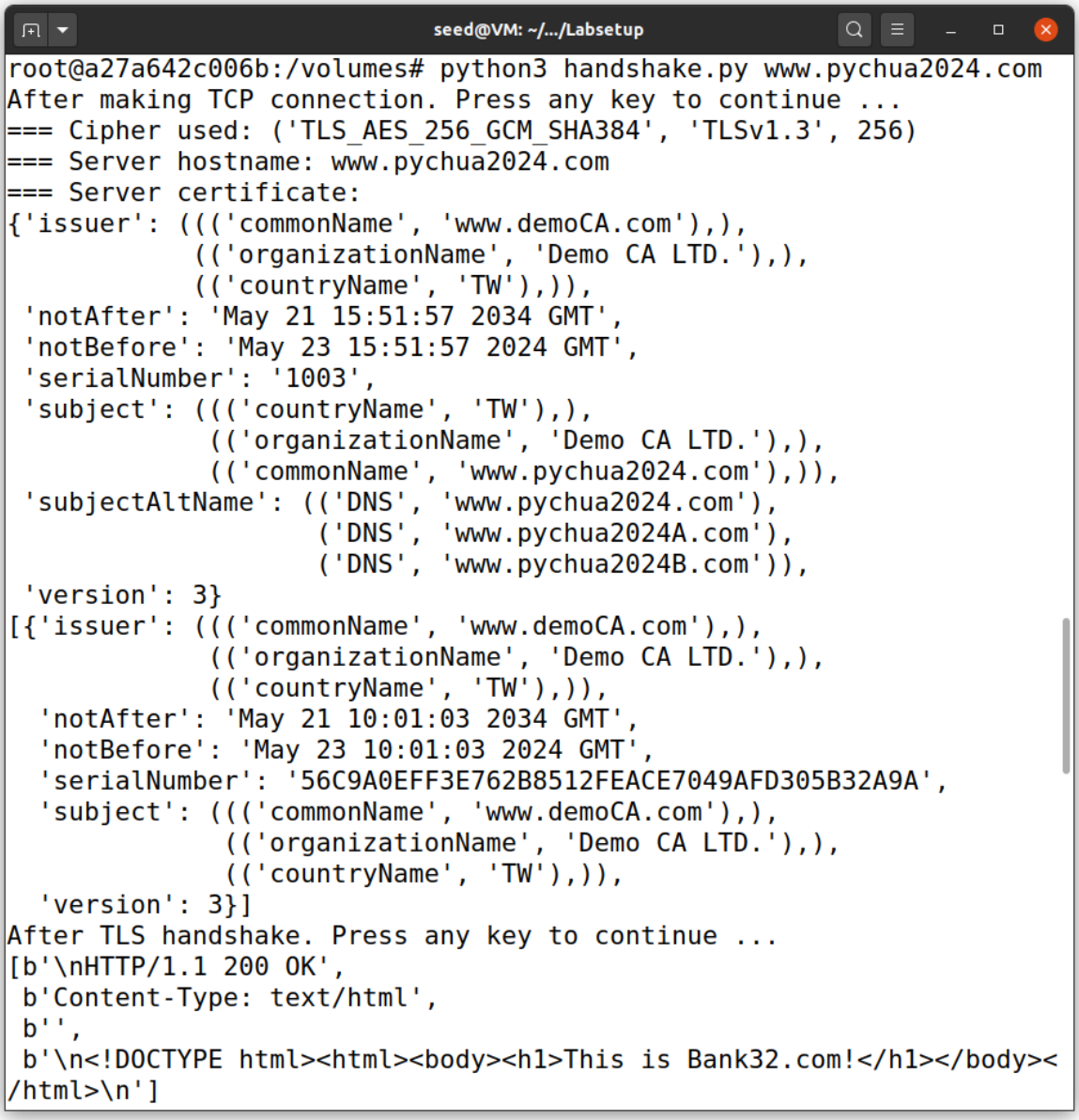


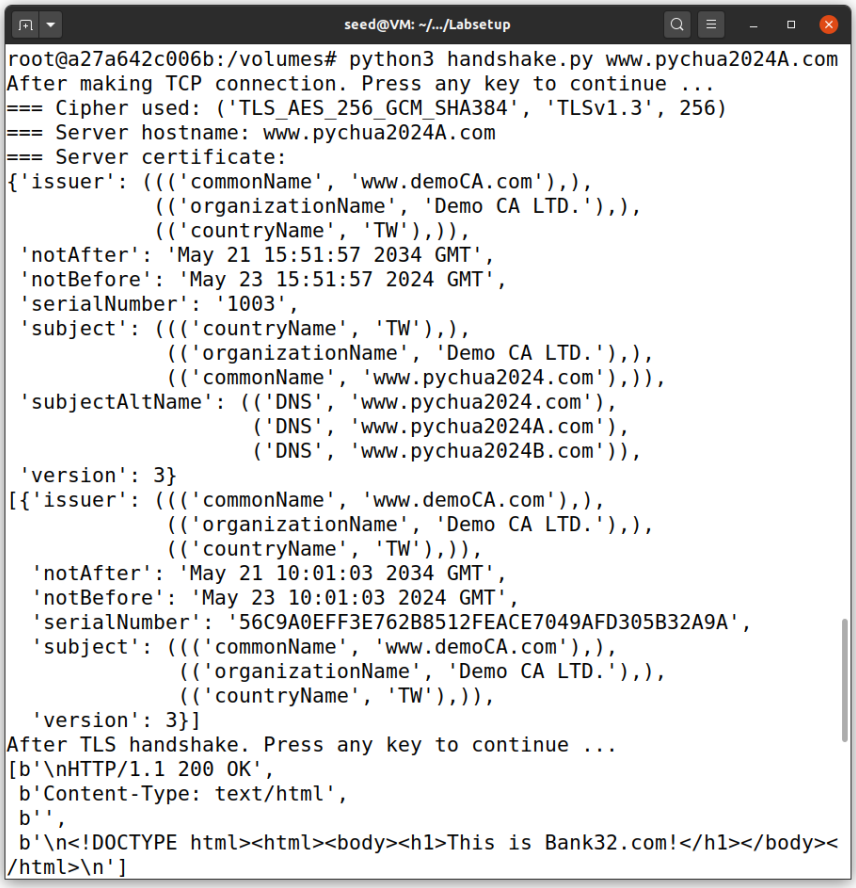
Task 2.3. Certificate with multiple names

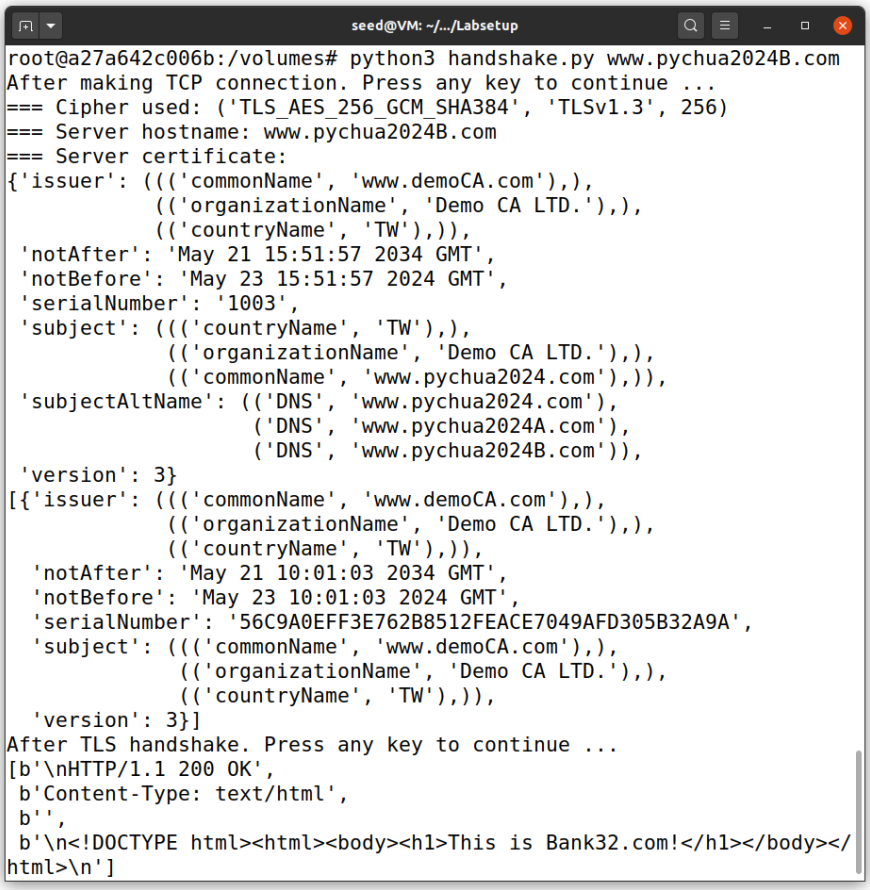
Server:



Client:

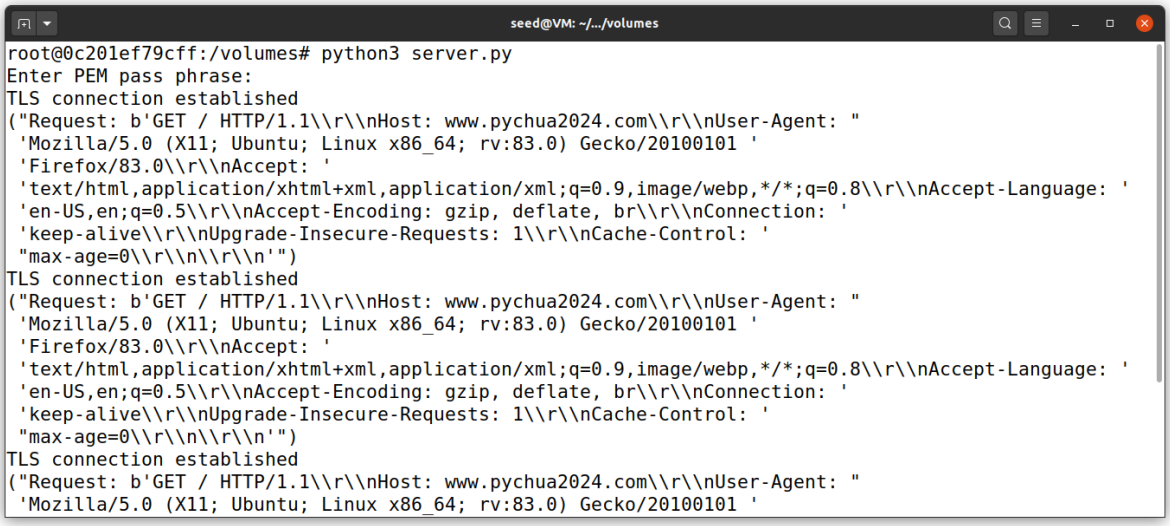

seed@VM: ~/.../Labsetup
root@a27a642c006b:/volumes# python3 handshake.py www.pychua2024.com
After making TCP connection. Press any key to continue ...
=== Cipher used: ('TLS_AES_256_GCM_SHA384', 'TLSv1.3', 256)
=== Server hostname: www.pychua2024.com
=== Server certificate:
{'issuer': ((('commonName', 'www.demoCA.com'),),
(('organizationName', 'Demo CA LTD.'),),
(('countryName', 'TW'),)),
'notAfter': 'May 21 15:51:57 2034 GMT',
'notBefore': 'May 23 15:51:57 2024 GMT',
'serialNumber': '1003',
'subject': ((('countryName', 'TW'),),
(('organizationName', 'Demo CA LTD.'),),
(('commonName', 'www.pychua2024.com'),)),
'subjectAltName': (('DNS', 'www.pychua2024.com'),
('DNS', 'www.pychua2024A.com'),
('DNS', 'www.pychua2024B.com')),
'version': 3}
[{'issuer': ((('commonName', 'www.demoCA.com'),),
(('organizationName', 'Demo CA LTD.'),),
(('countryName', 'TW'),)),
'notAfter': 'May 21 10:01:03 2034 GMT',
'notBefore': 'May 23 10:01:03 2024 GMT',
'serialNumber': '56C9A0EFF3E762B8512FEACE7049AFD305B32A9A',
'subject': ((('commonName', 'www.demoCA.com'),),
(('organizationName', 'Demo CA LTD.'),),
(('countryName', 'TW'),)),
'version': 3}]
After TLS handshake. Press any key to continue ...
[b'\nHTTP/1.1 200 OK',
b'Content-Type: text/html',
b'',
b'\n<!DOCTYPE html><html><body><h1>This is Bank32.com!</h1></body><
/html>\n']

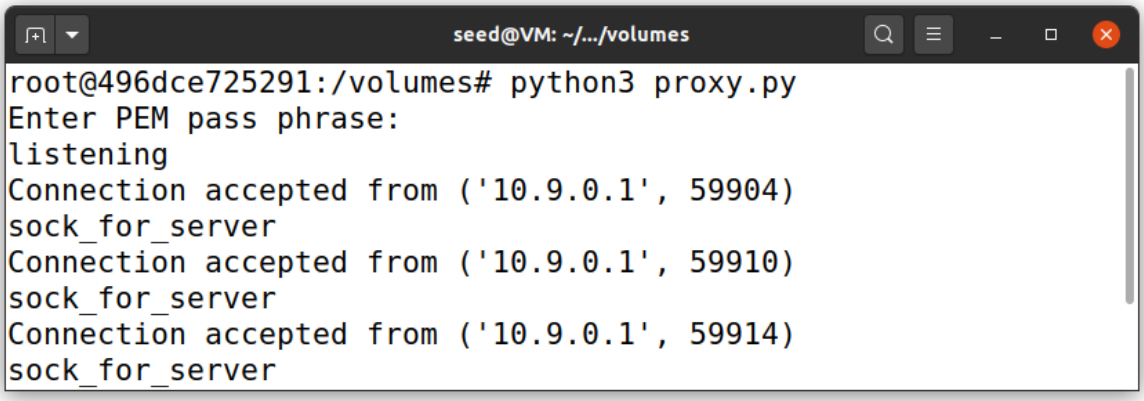

seed@VM: ~/.../Labsetup
root@a27a642c006b:/volumes# python3 handshake.py www.pychua2024A.com
After making TCP connection. Press any key to continue ...
=== Cipher used: ('TLS_AES_256_GCM_SHA384', 'TLSv1.3', 256)
=== Server hostname: www.pychua2024A.com
=== Server certificate:
{'issuer': ((('commonName', 'www.demoCA.com'),),
(('organizationName', 'Demo CA LTD.'),),
(('countryName', 'TW'),)),
'notAfter': 'May 21 15:51:57 2034 GMT',
'notBefore': 'May 23 15:51:57 2024 GMT',
'serialNumber': '1003',
'subject': ((('countryName', 'TW'),),
(('organizationName', 'Demo CA LTD.'),),
(('commonName', 'www.pychua2024.com'),)),
'subjectAltName': (('DNS', 'www.pychua2024.com'),
('DNS', 'www.pychua2024A.com'),
('DNS', 'www.pychua2024B.com')),
'version': 3}
[{'issuer': ((('commonName', 'www.demoCA.com'),),
(('organizationName', 'Demo CA LTD.'),),
(('countryName', 'TW'),)),
'notAfter': 'May 21 10:01:03 2034 GMT',
'notBefore': 'May 23 10:01:03 2024 GMT',
'serialNumber': '56C9A0EFF3E762B8512FEACE7049AFD305B32A9A',
'subject': ((('commonName', 'www.demoCA.com'),),
(('organizationName', 'Demo CA LTD.'),),
(('countryName', 'TW'),)),
'version': 3}]
After TLS handshake. Press any key to continue ...
[b'\nHTTP/1.1 200 OK',
b'Content-Type: text/html',
b'',
b'\n<!DOCTYPE html><html><body><h1>This is Bank32.com!</h1></body><
/html>\n']

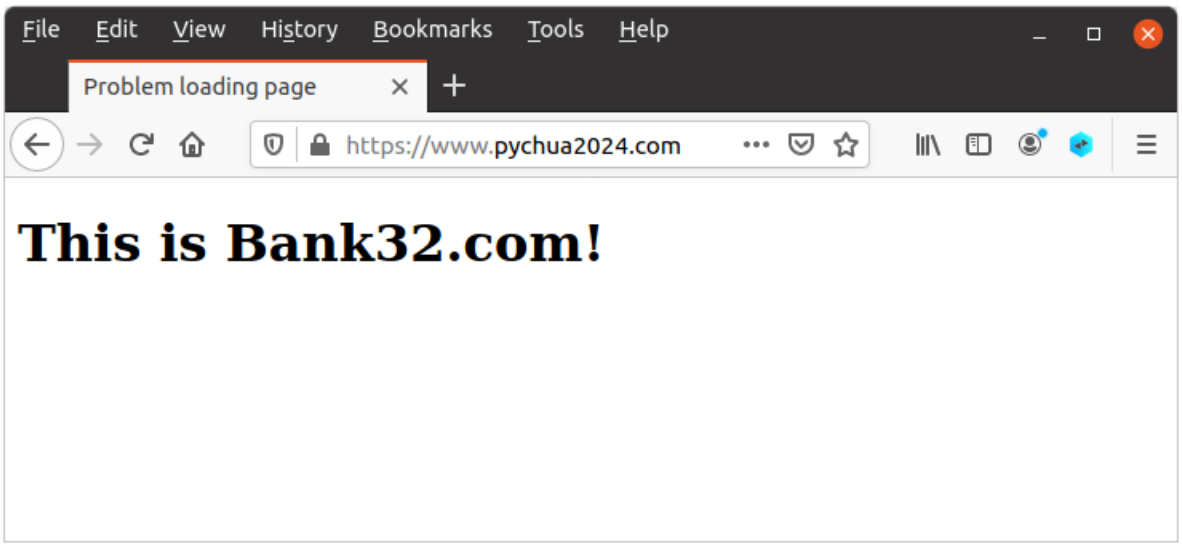

seed@VM: ~/.../Labsetup
root@a27a642c006b:/volumes# python3 handshake.py www.pychua2024B.com
After making TCP connection. Press any key to continue ...
=== Cipher used: ('TLS_AES_256_GCM_SHA384', 'TLSv1.3', 256)
=== Server hostname: www.pychua2024B.com
=== Server certificate:
{'issuer': ((('commonName', 'www.demoCA.com'),),
(('organizationName', 'Demo CA LTD.'),),
(('countryName', 'TW'),)),
'notAfter': 'May 21 15:51:57 2034 GMT',
'notBefore': 'May 23 15:51:57 2024 GMT',
'serialNumber': '1003',
'subject': ((('countryName', 'TW'),),
(('organizationName', 'Demo CA LTD.'),),
(('commonName', 'www.pychua2024.com'),)),
'subjectAltName': (('DNS', 'www.pychua2024.com'),
('DNS', 'www.pychua2024A.com'),
('DNS', 'www.pychua2024B.com')),
'version': 3}
[{'issuer': ((('commonName', 'www.demoCA.com'),),
(('organizationName', 'Demo CA LTD.'),),
(('countryName', 'TW'),)),
'notAfter': 'May 21 10:01:03 2034 GMT',
'notBefore': 'May 23 10:01:03 2024 GMT',
'serialNumber': '56C9A0EFF3E762B8512FEACE7049AFD305B32A9A',
'subject': ((('commonName', 'www.demoCA.com'),),
(('organizationName', 'Demo CA LTD.'),),
(('countryName', 'TW'),)),
'version': 3}]
After TLS handshake. Press any key to continue ...
[b'\nHTTP/1.1 200 OK',
b'Content-Type: text/html',
b'',
b'\n<!DOCTYPE html><html><body><h1>This is Bank32.com!</h1></body></
html>\n']

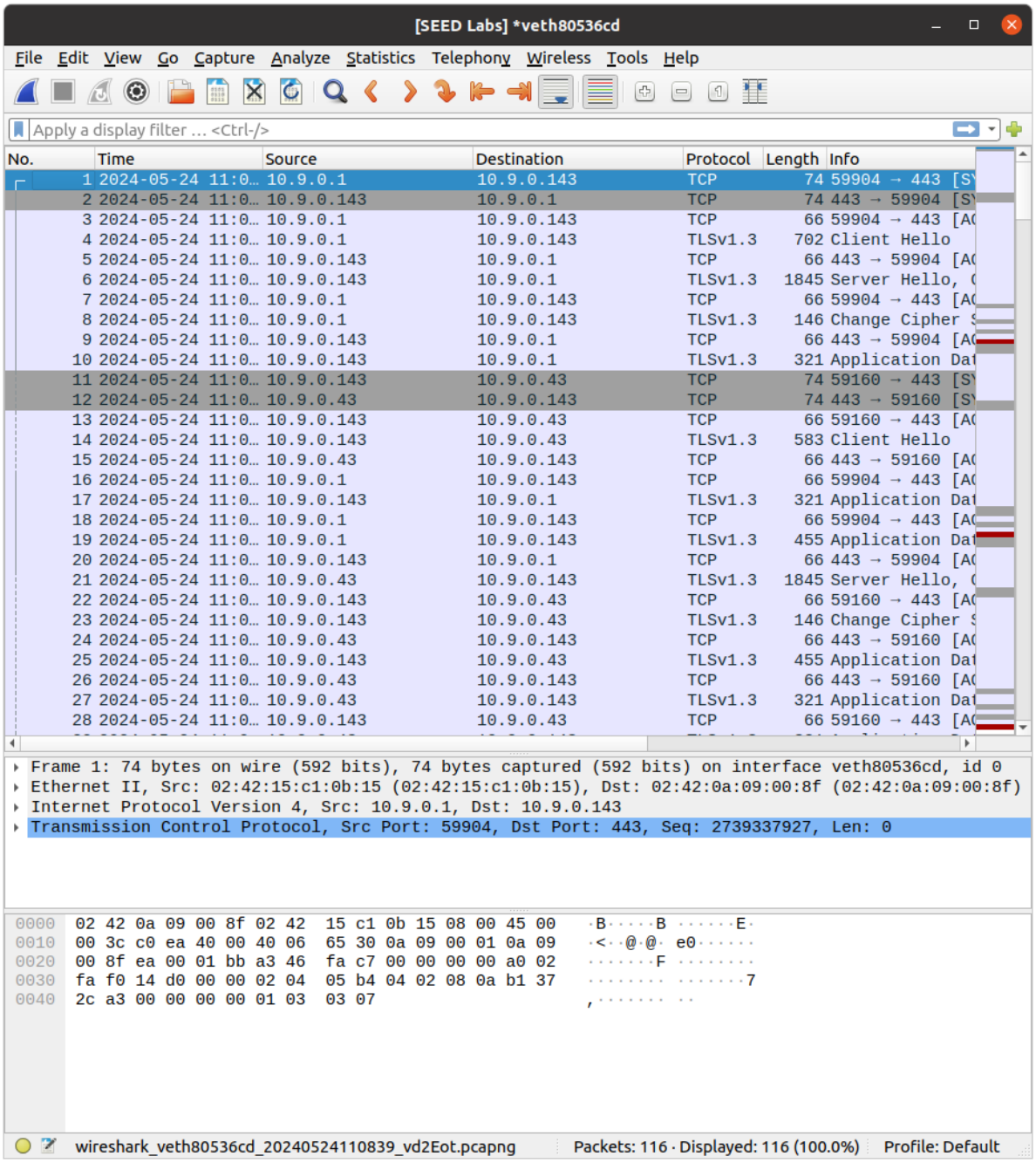
Task 3: A Simple HTTPS Proxy

MITM attack against own server:



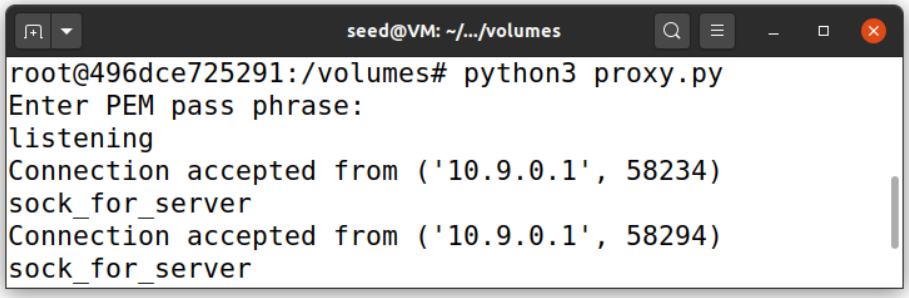


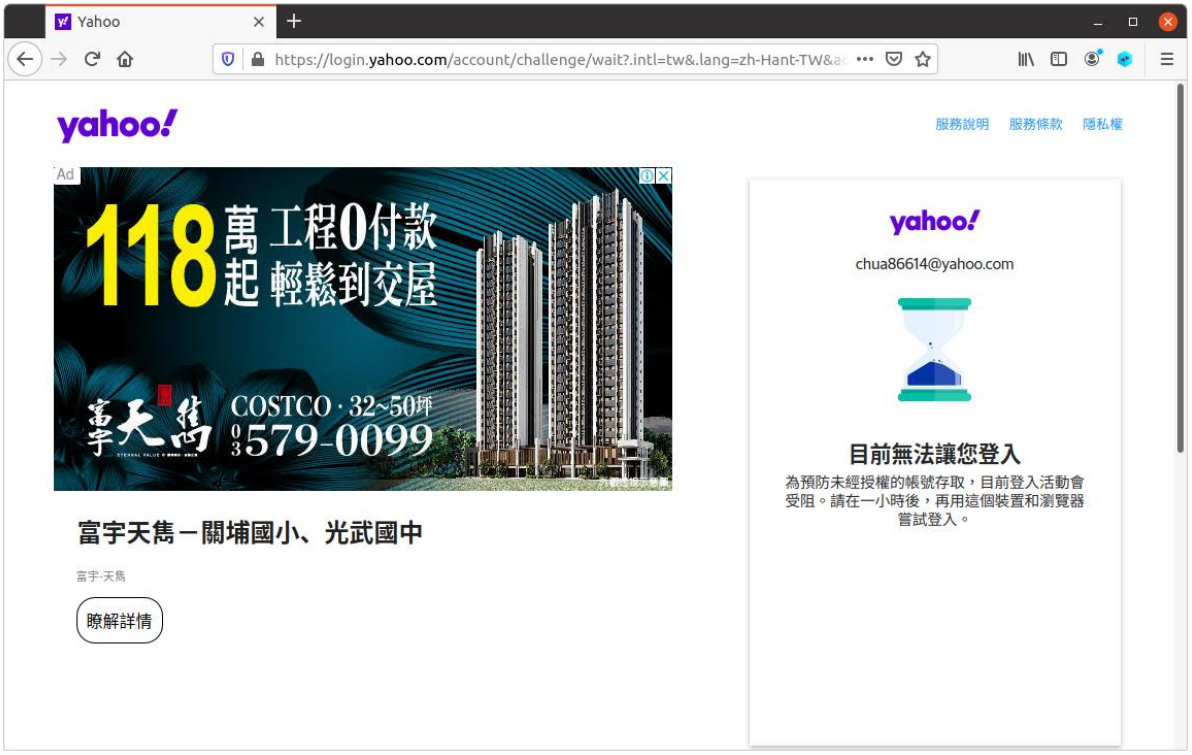



[SEED Labs] *veth80536cd
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
No. Time Source Destination Protocol Length Info
1 2024-05-24 11:0… 10.9.0.1 10.9.0.143 TCP 74 59904 → 443 [S
2 2024-05-24 11:0… 10.9.0.143 10.9.0.1 TCP 74 443 → 59904 [S
3 2024-05-24 11:0… 10.9.0.1 10.9.0.143 TCP 66 59904 → 443 [A
4 2024-05-24 11:0… 10.9.0.1 10.9.0.143 TLSv1.3 702 Client Hello
5 2024-05-24 11:0… 10.9.0.143 10.9.0.1 TCP 66 443 → 59904 [A
6 2024-05-24 11:0… 10.9.0.143 10.9.0.1 TLSv1.3 1845 Server Hello,
7 2024-05-24 11:0… 10.9.0.1 10.9.0.143 TCP 66 59904 → 443 [A
8 2024-05-24 11:0… 10.9.0.1 10.9.0.143 TLSv1.3 146 Change Cipher
9 2024-05-24 11:0… 10.9.0.143 10.9.0.1 TCP 66 443 → 59904 [A
10 2024-05-24 11:0… 10.9.0.143 10.9.0.1 TLSv1.3 321 Application Da
11 2024-05-24 11:0… 10.9.0.143 10.9.0.43 TCP 74 59160 → 443 [S
12 2024-05-24 11:0… 10.9.0.43 10.9.0.143 TCP 74 443 → 59160 [S
13 2024-05-24 11:0… 10.9.0.143 10.9.0.43 TCP 66 59160 → 443 [A
14 2024-05-24 11:0… 10.9.0.143 10.9.0.43 TLSv1.3 583 Client Hello
15 2024-05-24 11:0… 10.9.0.43 10.9.0.143 TCP 66 443 → 59160 [A
16 2024-05-24 11:0… 10.9.0.1 10.9.0.143 TCP 66 59904 → 443 [A
17 2024-05-24 11:0… 10.9.0.143 10.9.0.1 TLSv1.3 321 Application Da
18 2024-05-24 11:0… 10.9.0.1 10.9.0.143 TCP 66 59904 → 443 [A
19 2024-05-24 11:0… 10.9.0.1 10.9.0.143 TLSv1.3 455 Application Da
20 2024-05-24 11:0… 10.9.0.143 10.9.0.1 TCP 66 443 → 59904 [A
21 2024-05-24 11:0… 10.9.0.43 10.9.0.143 TLSv1.3 1845 Server Hello,
22 2024-05-24 11:0… 10.9.0.143 10.9.0.43 TCP 66 59160 → 443 [A
23 2024-05-24 11:0… 10.9.0.143 10.9.0.43 TLSv1.3 146 Change Cipher
24 2024-05-24 11:0… 10.9.0.43 10.9.0.143 TCP 66 443 → 59160 [A
25 2024-05-24 11:0… 10.9.0.143 10.9.0.43 TLSv1.3 455 Application Da
26 2024-05-24 11:0… 10.9.0.43 10.9.0.143 TCP 66 443 → 59160 [A
27 2024-05-24 11:0… 10.9.0.43 10.9.0.143 TLSv1.3 321 Application Da
28 2024-05-24 11:0… 10.9.0.143 10.9.0.43 TCP 66 59160 → 443 [A
Frame 1: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface veth80536cd, id 0
Ethernet II, Src: 02:42:15:c1:0b:15 (02:42:15:c1:0b:15), Dst: 02:42:0a:09:00:8f (02:42:0a:09:00:8f)
Internet Protocol Version 4, Src: 10.9.0.1, Dst: 10.9.0.143
Transmission Control Protocol, Src Port: 59904, Dst Port: 443, Seq: 2739337927, Len: 0
0000 02 42 0a 09 00 8f 02 42 15 c1 0b 15 08 00 45 00 ·B·····B ·····E·
0010 00 3c c0 ea 40 00 40 06 65 30 0a 09 00 01 0a 09 ·<··@·@· e0·····
0020 00 8f ea 00 01 bb a3 46 fa c7 00 00 00 00 a0 02 ·····F ·····
0030 fa f0 14 d0 00 00 02 04 05 b4 04 02 08 0a b1 37 ····· ·····7
0040 2c a3 00 00 00 00 01 03 03 07 ,····· ··
wireshark_veth80536cd_20240524110839_vd2Eot.pcapng
Packets: 116 · Displayed: 116 (100.0%)
Profile: Default

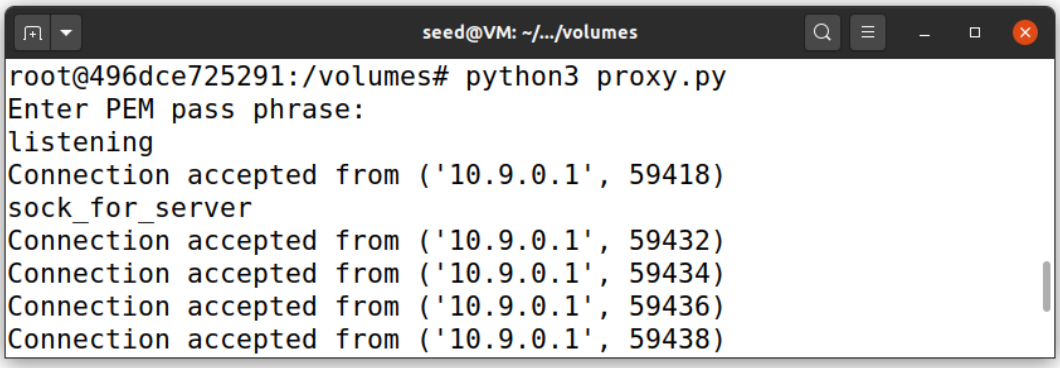
MITM attack on a real HTTPS website:

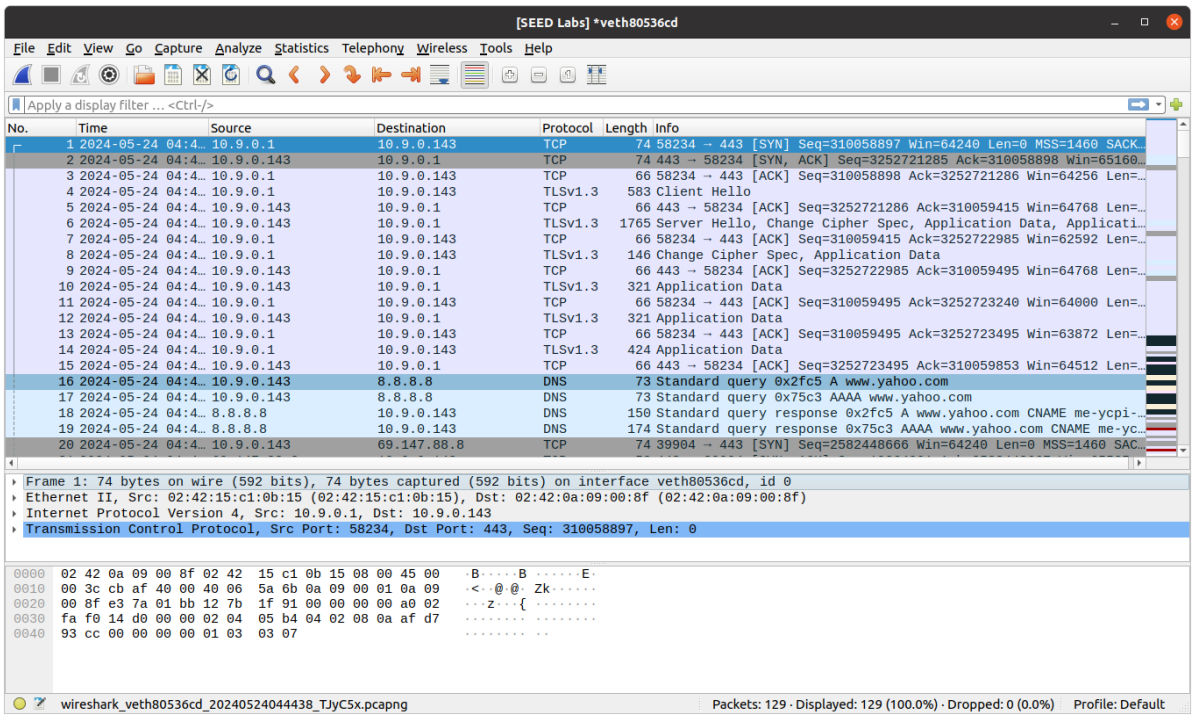
Originally tried with www.yahoo.com, however unable to login because of security concern





Hence, tried with codeforces.com





Codeforces

https://codeforces.com

CODEFORCES
Sponsored by TON

Enter | Register

HOME TOP CATALOG CONTESTS GYM PROBLEMSET GROUPS RATING EDU API CALENDAR HELP

ICPC Alumni Event in Sillicon Valley 2024

By **Temirulan**, history, 47 hours ago,

We invite you to first ICPC alumni event in Silicon Valley on June 4 at 6pm.

This is a unique event that will bring together top ICPC alumni and friends. The event will be attended by ICPC President Bill Poucher, Freedom Holding CEO Timur Turlov and Bagdat Mussin, President of the Kazakh Federation of Competitive Programming.

Register here: https://lu.ma/ICPC_Meetup

P.S: wdyt about some small entertainy contest?

Full text and comments »

icpc

+152 Temirulan 47 hours ago 22

Codeforces Round #946 (Div. 3)

By Vladosiya, history, 5 days ago, translation

→ Pay attention

Before contest
Codeforces Round (Div. 1 + Div. 2)
27:08:45
Register now »
*has extra registration

→ Streams

Round 1975 Div 1 + Div 2 Solution Discussion (with Jan)
By Shayan
Before stream 30:13:45

View all →

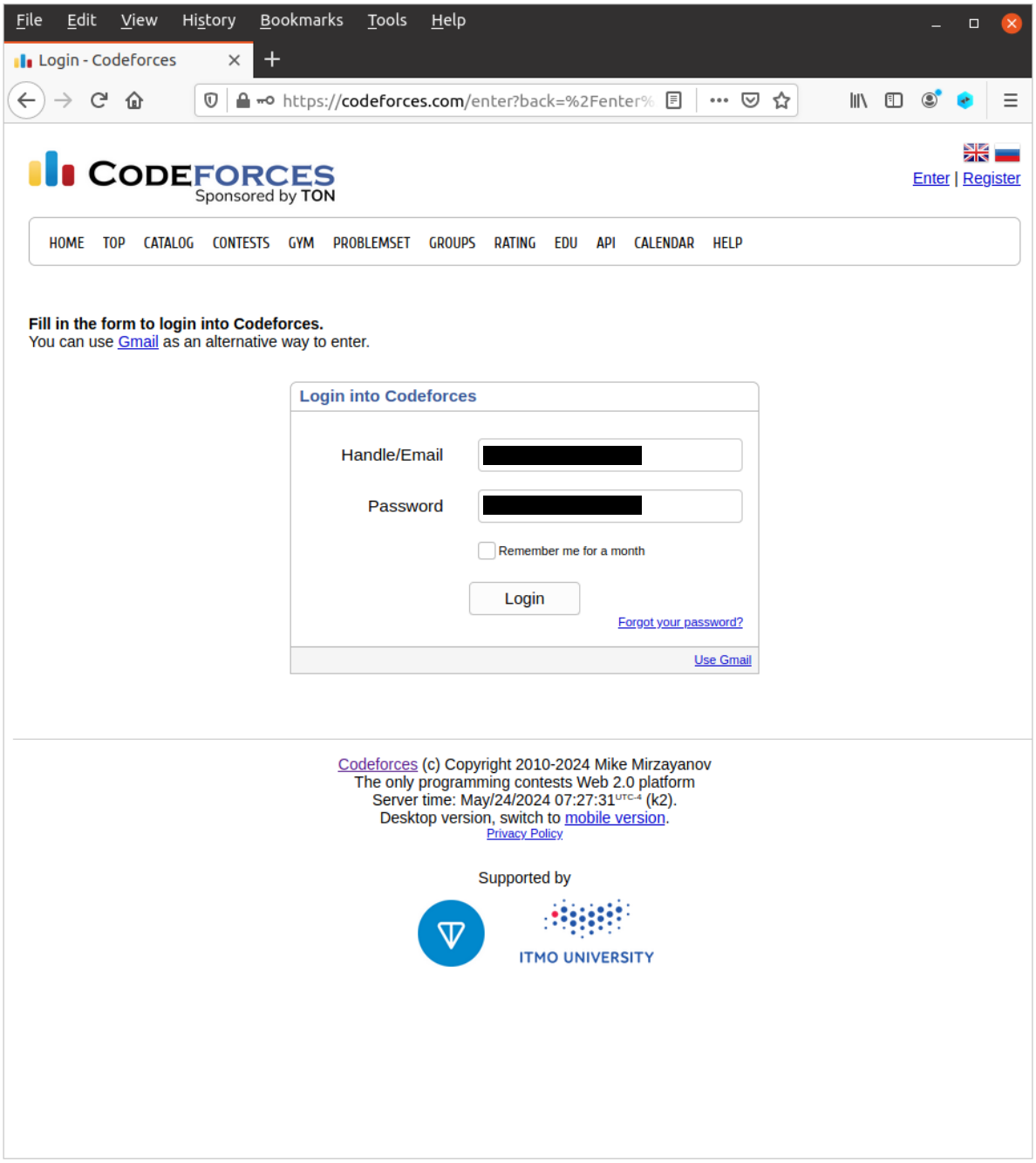
→ Top rated

#	User	Rating
1	tourist	3690
2	jiangly	3647
3	Benq	3581
4	orzdevinwang	3570
5	Geothermal	3569
5	cnnfls_csy	3569
7	Radewoosh	3509
8	ecnerwala	3486
9	jqdai0815	3474
10	gyh20	3447

Countries | Cities | Organizations View all →

→ Top contributors

#	User	Contrib.
1	maomao90	171
2	adamant	164
3	awoo	163


File Edit View History Bookmarks Tools Help
Login - Codeforces
https://codeforces.com/enter?back=%2Fenter%
CODEFORCES
Sponsored by TON
Enter | Register
HOME TOP CATALOG CONTESTS GYM PROBLEMSET GROUPS RATING EDU API CALENDAR HELP
Fill in the form to login into Codeforces.
You can use Gmail as an alternative way to enter.
Login into Codeforces
Handle/Email
Password
Remember me for a month
Login
Forgot your password?
Use Gmail
Codeforces (c) Copyright 2010-2024 Mike Mirzayanov
The only programming contests Web 2.0 platform
Server time: May/24/2024 07:27:31UTC-4 (k2).
Desktop version, switch to mobile version.
Privacy Policy
Supported by
ITMO UNIVERSITY