Exercise 2:

1. Connection to a Mail service (Gmail here) by accessing IMAP Server using openssl

```
salmane@salmane-VirtualBox:~$ openssl s_client -crlf -connect imap.gmail.com:993
CONNECTED(00000003)
depth=2 C = US, O = Google Trust Services LLC, CN = GTS Root R1
verify return:1
depth=1 C = US, O = Google Trust Services, CN = WR2
verify return:1
depth=0 CN = imap.gmail.com
verify return:1
---
Certificate chain
 0 s:CN = imap.gmail.com
   i:C = US, O = Google Trust Services, CN = WR2
   a:PKEY: id-ecPublicKey, 256 (bit); sigalg: RSA-SHA256
   v:NotBefore: Jun 13 16:28:56 2024 GMT; NotAfter: Sep  5 16:28:55 2024 GMT
 1 s:C = US, O = Google Trust Services, CN = WR2
   i:C = US, O = Google Trust Services LLC, CN = GTS Root R1
   a:PKEY: rsaEncryption, 2048 (bit); sigalg: RSA-SHA256
   v:NotBefore: Dec 13 09:00:00 2023 GMT; NotAfter: Feb 20 14:00:00 2029 GMT
 2 s:C = US, O = Google Trust Services LLC, CN = GTS Root R1
   i:C = BE, O = GlobalSign nv-sa, OU = Root CA, CN = GlobalSign Root CA
   a:PKEY: rsaEncryption, 4096 (bit); sigalg: RSA-SHA256
   v:NotBefore: Jun 19 00:00:42 2020 GMT; NotAfter: Jan 28 00:00:42 2028 GMT
---
Server certificate
-----BEGIN CERTIFICATE-----
MIIEVzCCAz+gAwIBAgIQTYpfMp7sYSgJgNTBlbLEvzANBgkqhkiG9w0BAQsFADA7
MQswCQYDVQQGEwJVUzEeMBwGA1UEChMVR29vZ2xlIFRydXN0IFNlcnZpY2VzMQww
CgYDVQQDEwNXUjIwHhcNMjQwNjEzMTYyODU2WhcNMjQwOTA1MTYyODU1WjAZMRcw
FQYDVQQDEw5pbWFwLmdtYWlsLmNvbTBZMBMGByqGSM49AgEGCCqGSM49AwEHA0IA
BNtAXMwYenRFaVnHezX2//5zA0Dg6s9WcUXycxOfnmIH7SXIjOqYEhtsscUF+d9R
atpwIPQI60rXn5bS5uEo+CGjggJCMIICPjAOBgNVHQ8BAf8EBAMCB4AwEwYDVR0l
BAwwCgYIKwYBBQUHAwEwDAYDVR0TAQH/BAIwADAdBgNVHQ4EFgQUGVHlG16waC8V
hQb+Mi3jZwiGFWQwHwYDVR0jBBgwFoAU3hse7XkV1D43JMMhu+w0OW1CsjAwWAYI
KwYBBQUHAQEETDBKMCEGCCsGAQUFBzABhhVodHRwOi8vby5wa2kuZ29vZy93cjIw
```

Login :

```
tag login mcpesalman@gmail.com uuzpcgrxjjyanjll
* CAPABILITY IMAP4rev1 UNSELECT IDLE NAMESPACE QUOTA ID XLIST CHILDREN X-GM-EXT-1 UIDPLUS (
ST-EXTENDED LIST-STATUS LITERAL- SPECIAL-USE APPENDLIMIT=35651584
tag OK mcpesalman@gmail.com authenticated (Success)
```

List Mailboxes:

```
tag LIST "" "*"
* LIST (\HasNoChildren) "/" "INBOX"
* LIST (\HasChildren \Noselect) "/" "[Gmail]"
* LIST (\All \HasNoChildren) "/" "[Gmail]/All Mail"
* LIST (\Drafts \HasNoChildren) "/" "[Gmail]/Drafts"
* LIST (\HasNoChildren \Important) "/" "[Gmail]/Important"
* LIST (\HasNoChildren \Sent) "/" "[Gmail]/Sent Mail"
* LIST (\HasNoChildren \Junk) "/" "[Gmail]/Spam"
* LIST (\Flagged \HasNoChildren) "/" "[Gmail]/Starred"
* LIST (\HasNoChildren \Trash) "/" "[Gmail]/Trash"
tag OK Success
```

Select INBOX from the list.

```
tag SELECT INBOX
* FLAGS (\Answered \Flagged \Draft \Deleted \Seen $NotPhishing $Phishing)
* OK [PERMANENTFLAGS (\Answered \Flagged \Draft \Deleted \Seen $NotPhishing $Phishing \*)] Flags permitted.
* OK [UIDVALIDITY 1] UIDs valid.
* 1523 EXISTS
* 0 RECENT
* OK [UIDNEXT 2802] Predicted next UID.
* OK [HIGHESTMODSEQ 434958]
tag OK [READ-WRITE] INBOX selected. (Success)
tag STATUS INBOX (MESSAGES)
* STATUS "INBOX" (MESSAGES 1523)
tag OK Success
```

Fetch the last received email.

```
tag FETCH 1523
tag BAD Could not parse command
tag fetch 1523 (BODY)
* 1523 FETCH (BODY (("TEXT" "PLAIN" ("CHARSET" "UTF-8") NIL NIL "QUOTED-PRINTABLE"
" 24987 500) "ALTERNATIVE"))
```

Logout:

```
tag LOGOUT
* BYE LOGOUT Requested
tag OK 73 good day (Success)
40F79E526C7F0000:error:0A000126:SSL routines:ssl3_read_n:unexpected eof while
salmane@salmane-VirtualBox:~$
```

2. Connect to a server using telnet:

```
salmane@salmane-VirtualBox:~$ telnet google.com 80
Trying 142.250.200.142...
Connected to google.com.
Escape character is '^]'.
```

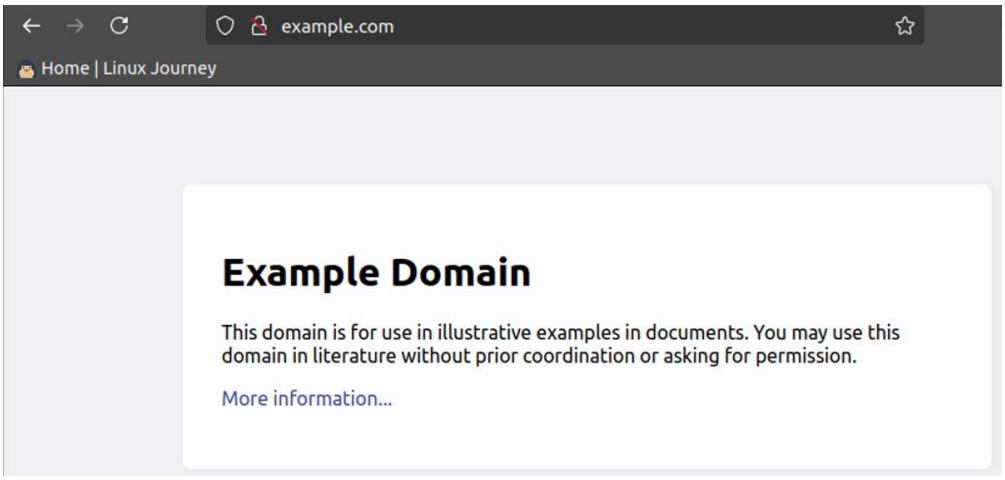
3. intercept an HTTP request coming to your browser using netcat

First, we install netcat using the command `*sudo apt install netcat*`

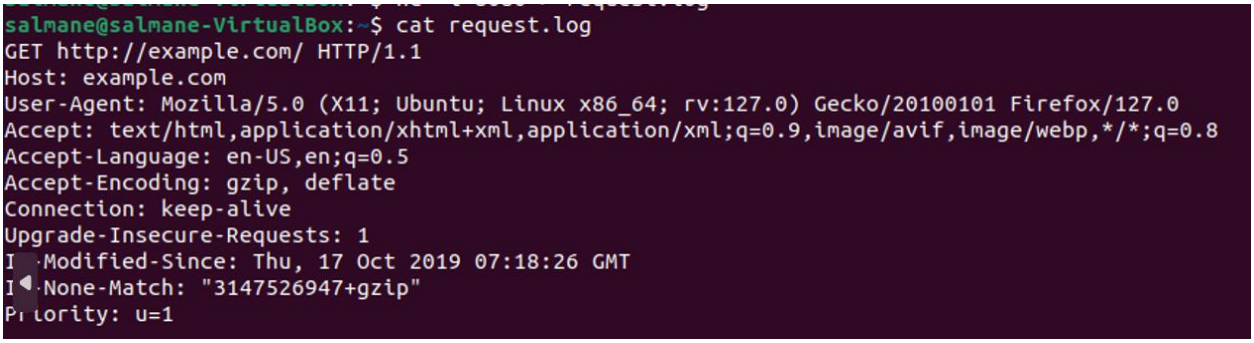
The we start listening to HTTP port which is 8080 and we will save the logs on a file (request.log)

```
salmane@salmane-VirtualBox:~$ nc -l 8080 > request.log
```

Now we need to perform an HTTP request (e.g. we will visit http://example.com)



And to make sure that we intercepted the request successfully we will check the logs file:



Therefore, we intercepted the HTTP request successfully !

4. Explain the difference between HTTP1 & HTTP2

HTTP/1.1 is a text-based protocol that processes one request per connection, leading to inefficiencies. HTTP/2 is a binary protocol that allows multiplexing, meaning multiple requests can be sent over a single connection simultaneously, improving performance.