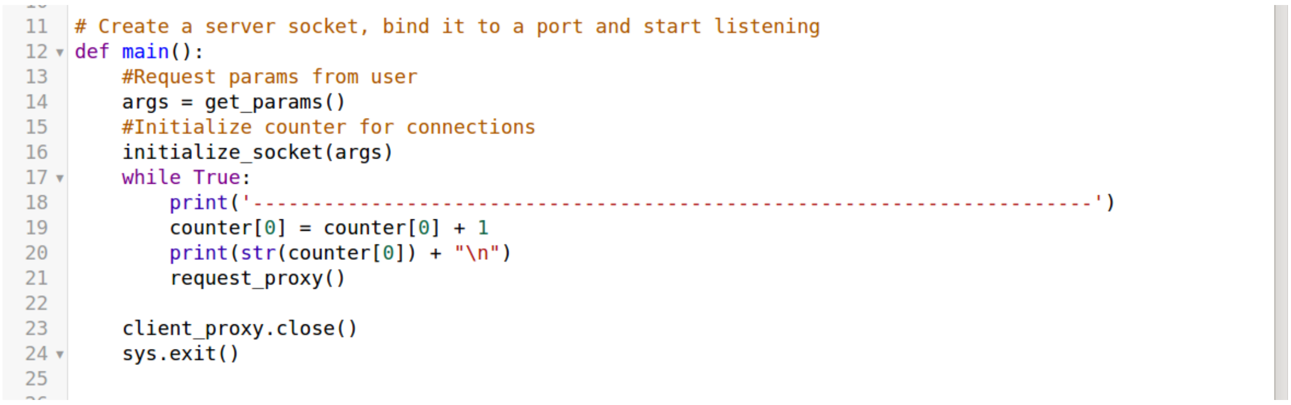
Julian Barnes
11/16/2017
Computer Networking
Project 3 Report

Introduction: Python 2.7, Ubuntu OS
Libraries: socket, sys, io, argparse, os, string, thread

Block Diagram

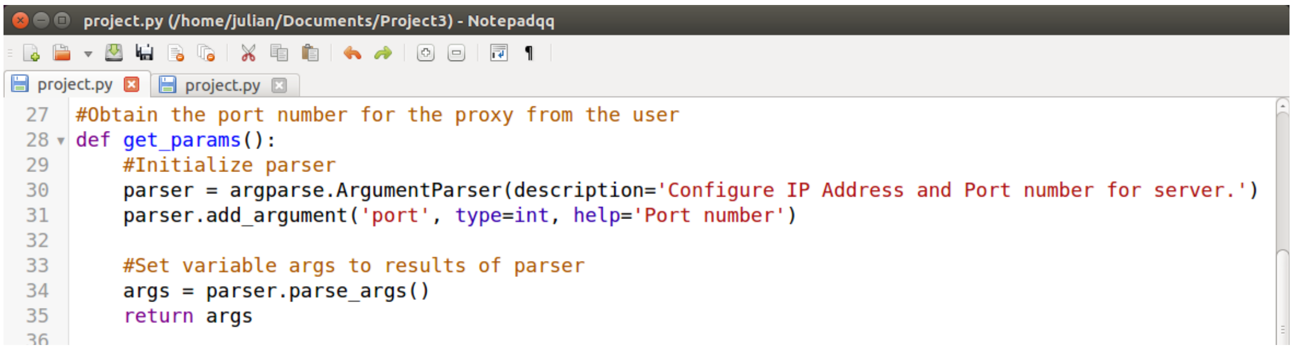
This main method first prompts the user for the port number he or she wants to establish the proxy on. Then it initializes the proxy with the given port number. A while loop is then started to continually receive requests from the client.



```
# Create a server socket, bind it to a port and start listening
def main():
    #Request params from user
    args = get_params()
    #Initialize counter for connections
    initialize_socket(args)
    while True:
        print('-----------------------------------------------------------------------')
        counter[0] = counter[0] + 1
        print(str(counter[0]) + "\n")
        request_proxy()

    client_proxy.close()
    sys.exit()
```

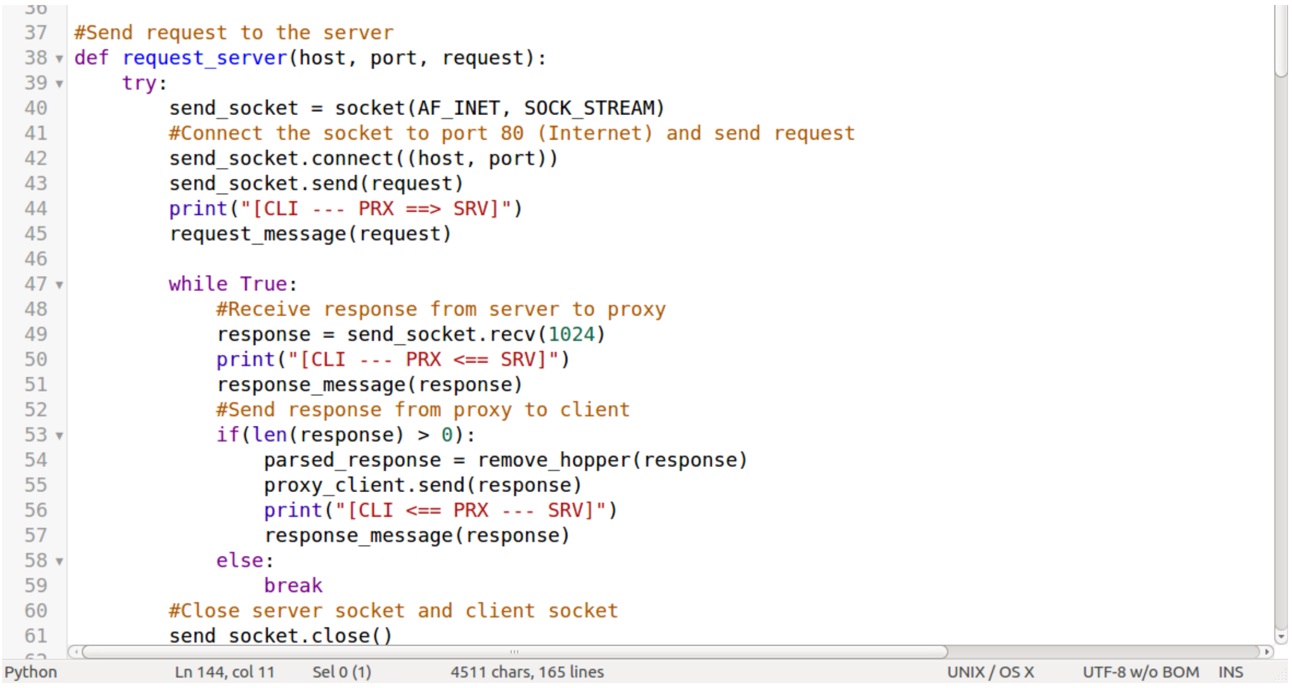
This method simply asks the user for a port number.



```
#Obtain the port number for the proxy from the user
def get_params():
    #Initialize parser
    parser = argparse.ArgumentParser(description='Configure IP Address and Port number for server.')
    parser.add_argument('port', type=int, help='Port number')

    #Set variable args to results of parser
    args = parser.parse_args()
    return args
```

This method forwards the request from the proxy to the server. It also starts a while loop that receives the server's response. Once the response is received by the proxy, the response is forwarded to the client

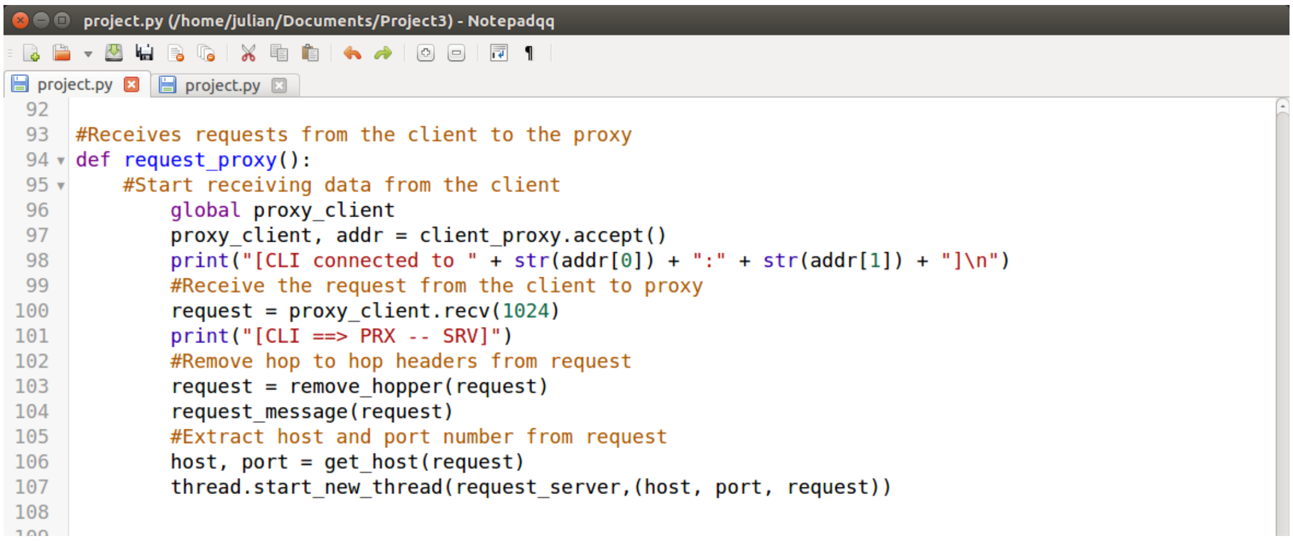


```
#Send request to the server
def request_server(host, port, request):
    try:
        send_socket = socket(AF_INET, SOCK_STREAM)
        #Connect the socket to port 80 (Internet) and send request
        send_socket.connect((host, port))
        send_socket.send(request)
        print("[CLI --- PRX ==> SRV]")
        request_message(request)

        while True:
            #Receive response from server to proxy
            response = send_socket.recv(1024)
            print("[CLI --- PRX <== SRV]")
            response_message(response)
            #Send response from proxy to client
            if(len(response) > 0):
                parsed_response = remove_hopper(response)
                proxy_client.send(response)
                print("[CLI <== PRX --- SRV]")
                response_message(response)
            else:
                break
        #Close server socket and client socket
        send_socket.close()
```

```
Python    Ln 144, col 11    Sel 0 (1)    4511 chars, 165 lines    UNIX / OS X    UTF-8 w/o BOM    INS
```

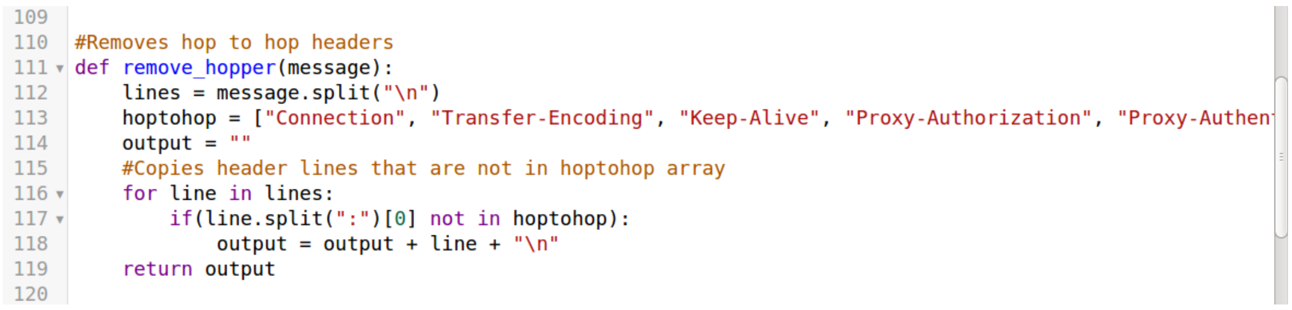
This method allows th proxy to receive requests from the client after the proxy accepts the client's connection request. A new thread for the request_server method is created every time this method runs. This allow the machine to handle multiple requests simultaneously. Thus, speeding up rendering time



project.py (/home/julian/Documents/Project3) - Notepadqq

```
#Receives requests from the client to the proxy
def request_proxy():
    #Start receiving data from the client
        global proxy_client
        proxy_client, addr = client_proxy.accept()
        print("[CLI connected to " + str(addr[0]) + ":" + str(addr[1]) + "]\n")
        #Receive the request from the client to proxy
        request = proxy_client.recv(1024)
        print("[CLI ==> PRX -- SRV]")
        #Remove hop to hop headers from request
        request = remove_hopper(request)
        request_message(request)
        #Extract host and port number from request
        host, port = get_host(request)
        thread.start_new_thread(request_server,(host, port, request))
```

When transferring requests from client to proxy to server, "hop to hop" headers must be removed. This method accomplishes that task.

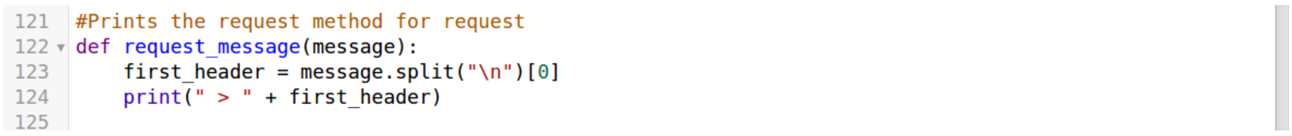


```

#Removes hop to hop headers
def remove_hopper(message):
    lines = message.split("\n")
    hoptohop = ["Connection", "Transfer-Encoding", "Keep-Alive", "Proxy-Authorization", "Proxy-Authen
    output = ""
    #Copies header lines that are not in hoptohop array
    for line in lines:
        if(line.split(":")[0] not in hoptohop):
            output = output + line + "\n"
    return output

```

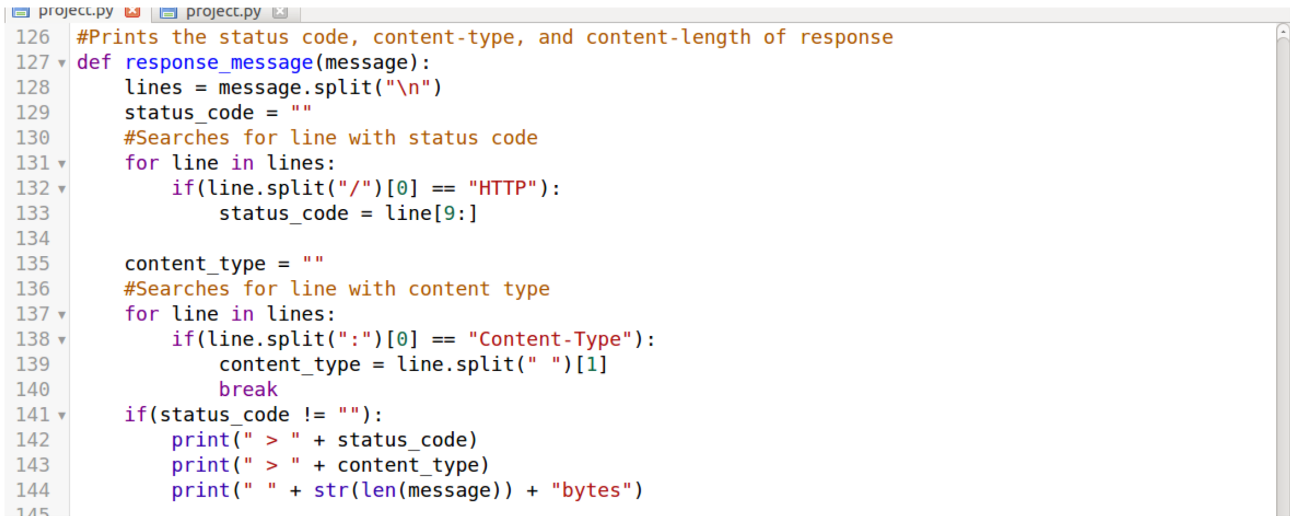
The request_message method parses an http request and prints out the first header of the message for logging.



```
#Prints the request method for request
def request_message(message):
    first_header = message.split("\n")[0]
    print(" > " + first_header)

```

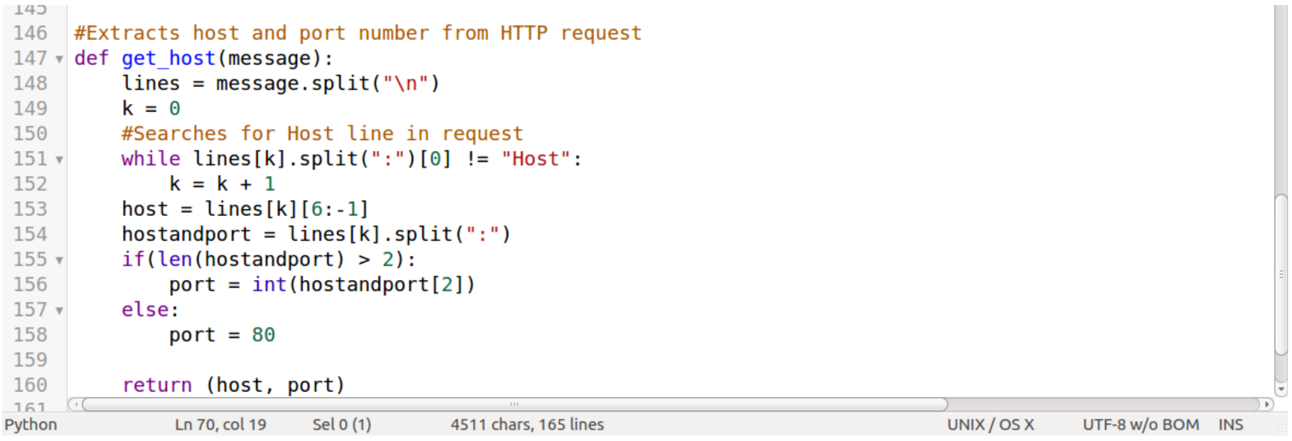
The response_message parses the http response and prints out the status code, the content type, and the content length for logging.



```
#Prints the status code, content-type, and content-length of response
def response_message(message):
    lines = message.split("\n")
    status_code = ""
    #Searches for line with status code
    for line in lines:
        if(line.split("/")[0] == "HTTP"):
            status_code = line[9:]

    content_type = ""
    #Searches for line with content type
    for line in lines:
        if(line.split(":")[0] == "Content-Type"):
            content_type = line.split(" ")[1]
            break
    if(status_code != ""):
        print(" > " + status_code)
        print(" > " + content_type)
        print("  " + str(len(message)) + "bytes")
```

This is a convenient method that extracts the host and port number from a http request. The results of this method are used in the request_server method.

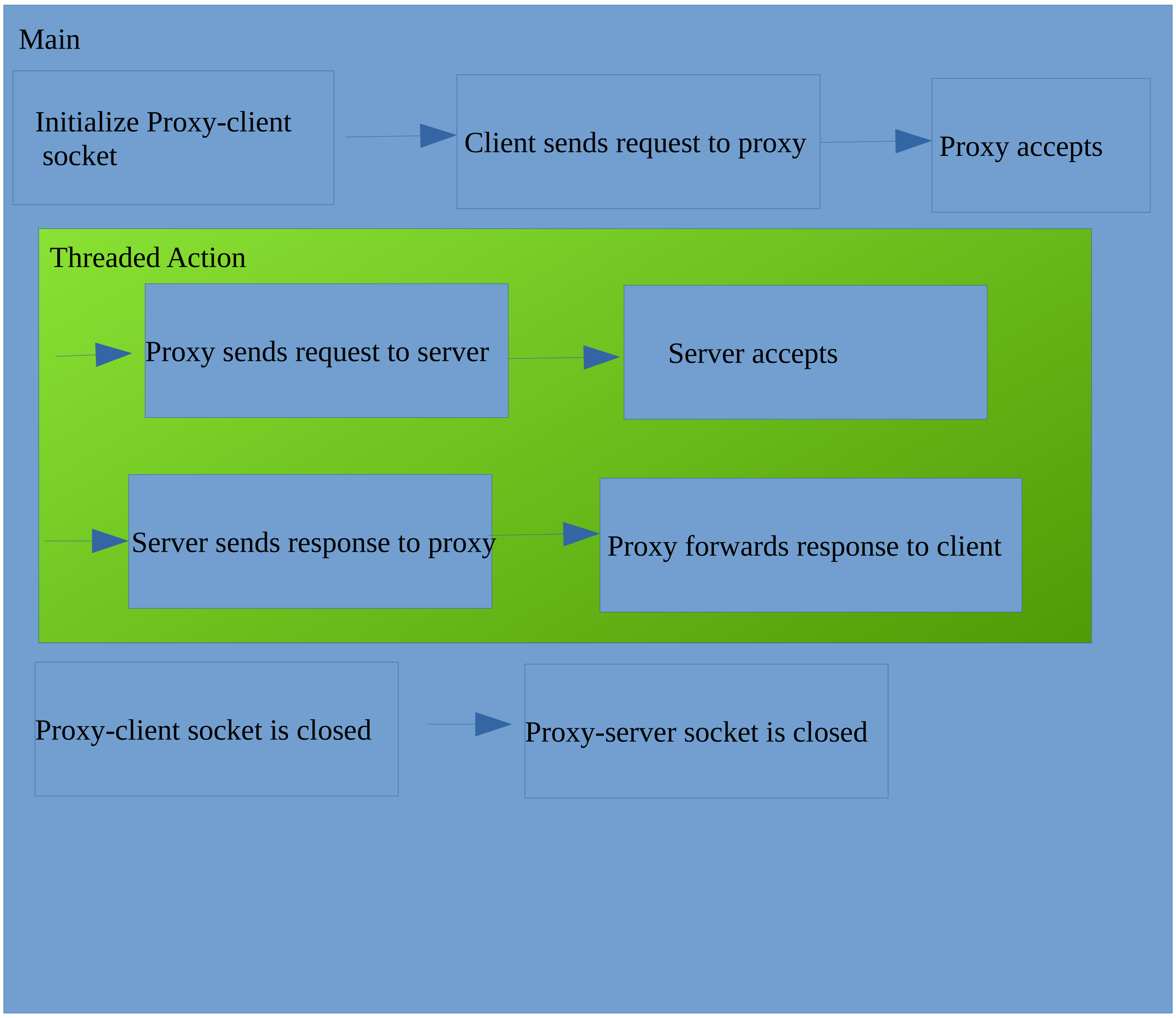


```
#Extracts host and port number from HTTP request
def get_host(message):
    lines = message.split("\n")
    k = 0
    #Searches for Host line in request
    while lines[k].split(":")[0] != "Host":
        k = k + 1
    host = lines[k][6:-1]
    hostandport = lines[k].split(":")
    if(len(hostandport) > 2):
        port = int(hostandport[2])
    else:
        port = 80

    return (host, port)
```

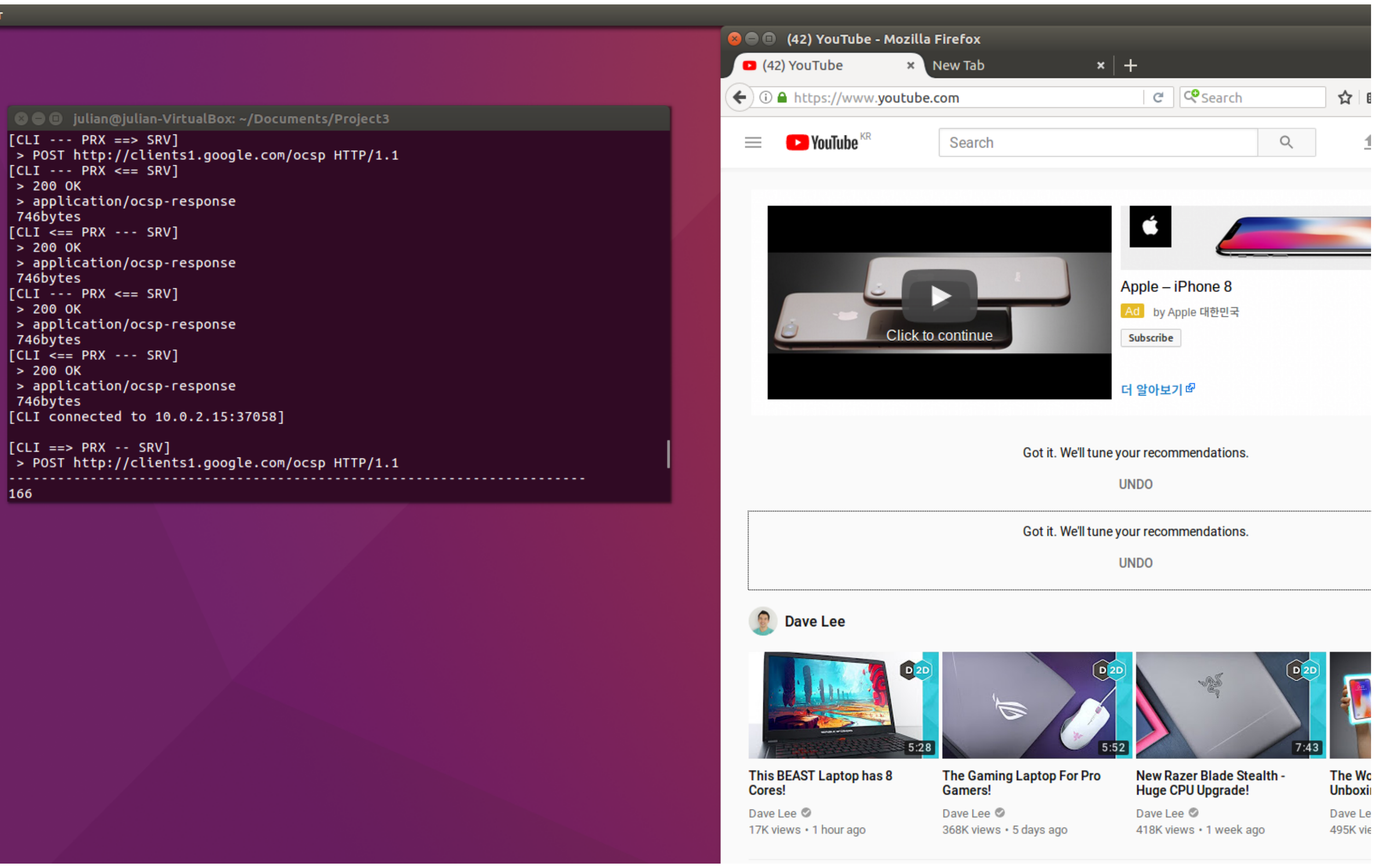
Python Ln 70, col 19 Sel 0 (1) 4511 chars, 165 lines UNIX / OS X UTF-8 w/o BOM INS

Flow Chart



Main
Initialize Proxy-client socket
Client sends request to proxy
Proxy accepts
Threaded Action
Proxy sends request to server
Server accepts
Server sends response to proxy
Proxy forwards response to client
Proxy-client socket is closed
Proxy-server socket is closed

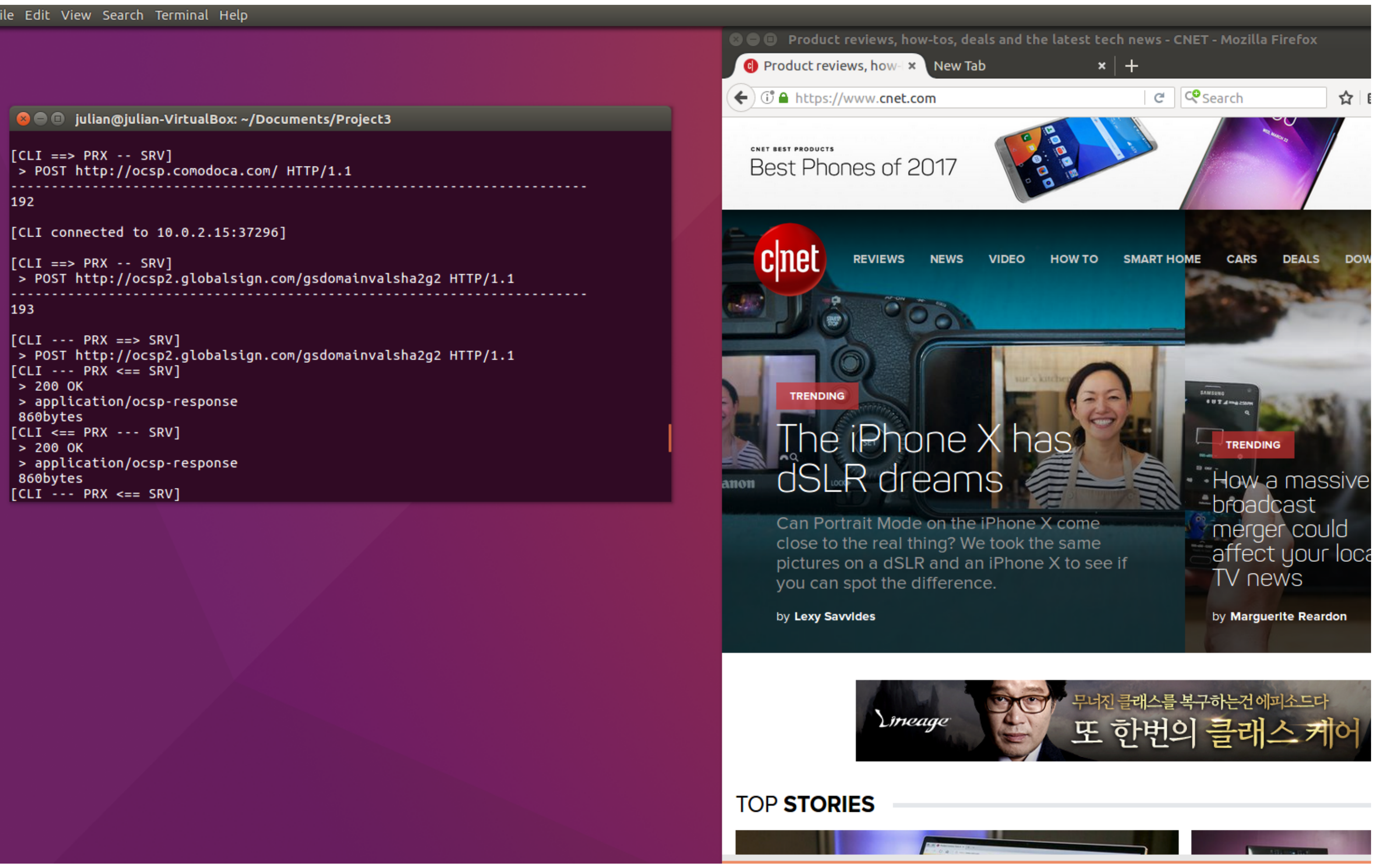
Examples



```
[CLI --- PRX ==> SRV]
 > POST http://clients1.google.com/ocsp HTTP/1.1
[CLI --- PRX <== SRV]
 > 200 OK
 > application/ocsp-response
 746bytes
[CLI <== PRX --- SRV]
 > 200 OK
 > application/ocsp-response
 746bytes
[CLI --- PRX <== SRV]
 > 200 OK
 > application/ocsp-response
 746bytes
[CLI <== PRX --- SRV]
 > 200 OK
 > application/ocsp-response
 746bytes
[CLI connected to 10.0.2.15:37058]

[CLI ==> PRX -- SRV]
 > POST http://clients1.google.com/ocsp HTTP/1.1
--------------------------------------------------------------------------
166
```

julian@julian-VirtualBox: ~/Documents/Project3
(42) YouTube - Mozilla Firefox
(42) YouTube
New Tab
https://www.youtube.com
Search
YouTube KR
Click to continue
Apple – iPhone 8
Ad by Apple 대한민국
Subscribe
더 알아보기
Got it. We'll tune your recommendations.
UNDO
Got it. We'll tune your recommendations.
UNDO
Dave Lee
This BEAST Laptop has 8 Cores!
Dave Lee
17K views • 1 hour ago
The Gaming Laptop For Pro Gamers!
Dave Lee
368K views • 5 days ago
New Razer Blade Stealth - Huge CPU Upgrade!
Dave Lee
418K views • 1 week ago



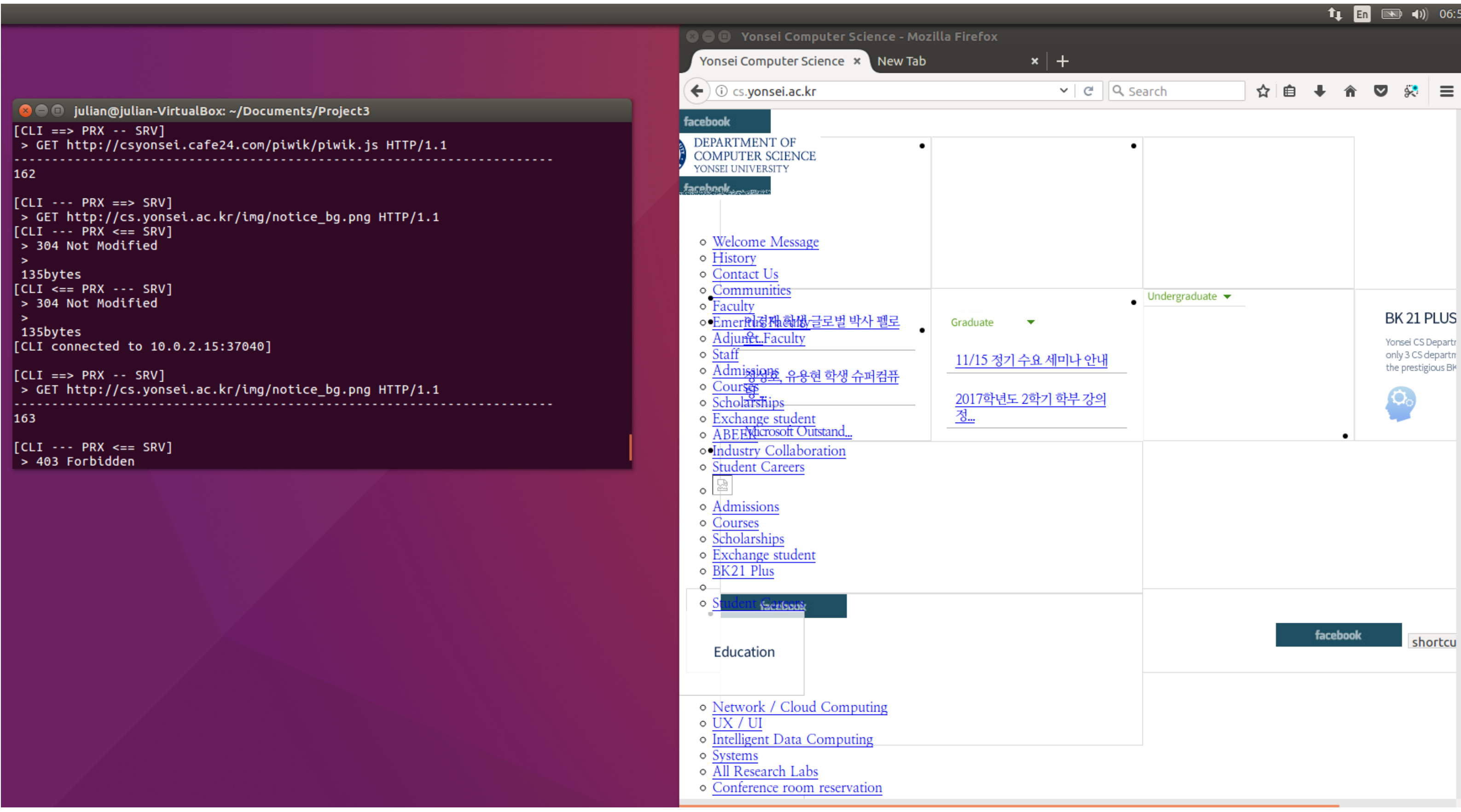
```
[CLI ==> PRX -- SRV]
 > POST http://ocsp.comodoca.com/ HTTP/1.1
--------------------------------------------------------------------------
192

[CLI connected to 10.0.2.15:37296]

[CLI ==> PRX -- SRV]
 > POST http://ocsp2.globalsign.com/gsdomainvalsha2g2 HTTP/1.1
--------------------------------------------------------------------------
193

[CLI --- PRX ==> SRV]
 > POST http://ocsp2.globalsign.com/gsdomainvalsha2g2 HTTP/1.1
[CLI --- PRX <== SRV]
 > 200 OK
 > application/ocsp-response
 860bytes
[CLI <== PRX --- SRV]
 > 200 OK
 > application/ocsp-response
 860bytes
[CLI --- PRX <== SRV]
```

ile Edit View Search Terminal Help
julian@julian-VirtualBox: ~/Documents/Project3
Product reviews, how-tos, deals and the latest tech news - CNET - Mozilla Firefox
Product reviews, how
New Tab
https://www.cnet.com
Search
CNET BEST PRODUCTS
Best Phones of 2017
cnet
REVIEWS NEWS VIDEO HOW TO SMART HOME CARS DEALS
TRENDING
The iPhone X has dSLR dreams
Can Portrait Mode on the iPhone X come close to the real thing? We took the same pictures on a dSLR and an iPhone X to see if you can spot the difference.
by Lexy Savvides
TRENDING
How a massive broadcast merger could affect your loca TV news
by Marguerite Reardon
Lineage
무너진 클래스를 복구하는건 에피소드다
또 한번의 클래스 케어
TOP STORIES



```
[CLI ==> PRX -- SRV]
 > GET http://csyonsei.cafe24.com/piwik/piwik.js HTTP/1.1
-------------------------------------------------------------------------
162

[CLI --- PRX ==> SRV]
 > GET http://cs.yonsei.ac.kr/img/notice_bg.png HTTP/1.1
[CLI --- PRX <== SRV]
 > 304 Not Modified
 >
 135bytes
[CLI <== PRX --- SRV]
 > 304 Not Modified
 >
 135bytes
[CLI connected to 10.0.2.15:37040]

[CLI ==> PRX -- SRV]
 > GET http://cs.yonsei.ac.kr/img/notice_bg.png HTTP/1.1
-------------------------------------------------------------------------
163

[CLI --- PRX <== SRV]
 > 403 Forbidden
```

Reference:
https://media.pearsoncmg.com/intl/ge/2017/cws/ema_ge_kurose_compnetwork_7/cw/index.php
http://www.freesoft.org/CIE/RFC/2068/143.html
https://www.mnot.net/blog/2011/07/11/what_proxies_must_do
https://www.tutorialspoint.com/python/python_multithreading.htm
https://null-byte.wonderhowto.com/how-to/sploit-make-proxy-server-python-0161232/
https://stackoverflow.com/questions/3475251/split-a-string-by-a-delimiter-in-python
https://www.ntu.edu.sg/home/ehchua/programming/webprogramming/HTTP_Basics.html