6/12/2023

Networks project documentation

Team members

Ziad ashraf ahmed ahmed 19p7095

Yassin Khaled Mostafa

Gannah allah Mohamed gaber

# The design of the system

Diagram

Description automatically generated

# The proxy.py code

import socket

import requests

import sys

import os

if len(sys.argv) <= 1:

print ('Usage : "python ProxyServer.py server\_ip"\n[server\_ip : It is the IP Address Of Proxy Server')

# Create a server socket, bind it to a port and start listening

tcpSerSock = socket.socket(socket.AF\_INET,socket.SOCK\_STREAM)

#fill in start

tcpSerSock.setsockopt(socket.SOL\_SOCKET, socket.SO\_REUSEADDR, 1)

tcpSerSock.bind(('localhost',8888))

tcpSerSock.listen(2)

# Fill in end.

while 1:

# Start receiving data from the client

print ('\n\nReady to serve...')

tcpCliSock, addr = tcpSerSock.accept() #return address and tcp client socket

print ('Received a connection from:', addr)

#fill in start

message = tcpCliSock.recv(4096)

#fill in end

if message == "":

continue

print (message)

# Extract the filename from the given message

file = message.split()[1]

filename = file.split('/')[1]

fileExist = "false"

filetouse = file

#////////////////Requirment 3/////////////////////

flag = -1

urlfile = open("URL\_BLOCKED.txt")

for i in urlfile:

if filename == i:

flag = 0

break

urlfile.close()

print( flag )

Blockedfile = open("Blocked.txt")

if flag == 0:

tcpCliSock.sendall("HTTP/1.0 403 Forbidden\r\n".encode()) # mod

tcpCliSock.sendall("Content-Type:text/html\r\n".encode()) # mod

tcpCliSock.sendall(Blockedfile.read().encode())

continue

Blockedfile.close()

#/////////////////////////////

try:

# Check wether the file exist in the cache

response = requests.get("http://" + filename)

print(response.status\_code)

if os.path.exists(filename):

if (response.status\_code != 200):

print ("in the if condition")

os.remove(filename)

raise IOError

f = open(filetouse[1:], "rb")#mod

outputdata = f.read() #was readlines-> in order to be not tuple

fileExist = "true"

# ProxyServer finds a cache hit and generates a response message

tcpCliSock.sendall("HTTP/1.0 200 OK\r\n".encode()) #mod

tcpCliSock.sendall("Content-Type:text/html\r\n".encode()) # mod

#fill in start

tcpCliSock.sendall("Content-Type: image/jpeg\r\n".encode())

tcpCliSock.sendall(outputdata)

f.close()

#fill in end

print ('Read from cache')

#return outputdata

# Error handling for file not found in cache

except IOError:

if fileExist == "false":

# Create a socket on the proxyserver

#fill in start

c = socket.socket(socket.AF\_INET,socket. SOCK\_STREAM)

#fill in end

file = file[1:]

hostn = file

hostn = file.replace("www.","",1)

try:

#fill in start

fileobj = c.makefile('rwb',0)

# Connect to the socket to port 80

port=80

if not "Referer" in message:

print("connecting to the web server ...")

c.connect((hostn, 80))

conneted=hostn

fileobj.write(b'GET / HTTP/1.0\r\n\r\n') # sent to browser server

else:

print("want to get the path in the referer: " + hostn)

c.connect((conneted, 80))

fileobj.write(b'GET /' + hostn + ' HTTP/1.0\r\n\r\n'.encode()) #sent to browser server

# fill in end

# check if it needs to be encoded

#fill in start

responseBuffer = fileobj.read()

print("response buffer printed")

# Create a new file in the cache for the requested file.

# Also send the response in the buffer to client socket and the corresponding file in the cache

tmpFile = open("./" + filename,"wb")

for i in range(0, len(responseBuffer)):

tmpFile.write(responseBuffer[i])

print("response buffer stored to file")

tcpCliSock.sendall("HTTP/1.0 200 OK\r\n".encode()) # mod

tcpCliSock.sendall("Content-Type:text/html\r\n".encode()) # mod

tcpCliSock.sendall("Content-Type: image/jpeg\r\n".encode())

tcpCliSock.sendall(responseBuffer)

print(responseBuffer)

print("responce buffer sent to client")

tmpFile.close()

# Fill in end.

#/////////////////requirement-1//////////////

except socket.gaierror:

print("error 404")

ERRORFile = open("Error.txt")

tcpCliSock.sendall("HTTP/1.0 404 page not found\r\n".encode()) # mod

tcpCliSock.sendall("Content-Type:text/html\r\n".encode()) # mod

tcpCliSock.sendall(ERRORFile.read().encode())

#//////////////////////////////////////////

except Exception as e:

print ("Illegal request")

print (e.args)

else:

# HTTP response message for file not found

#fill in start

print("error 404")

ERRORFile = open("Error.txt")

tcpCliSock.sendall("HTTP/1.0 404 page not found\r\n".encode()) # mod

tcpCliSock.sendall("Content-Type:text/html\r\n".encode()) # mod

tcpCliSock.sendall(ERRORFile.read().encode())

#fill in end

tcpCliSock.close()

# Fill in start.

tcpSerSock.flush()

tcpSerSock.close()

# Fill in end.-

# The screenshots

Step 1: write <http://localhost:portnumber/website> for the first time(cache now is empty)

The proxy server will receive the request form the browser and forward it to the website server then receive the response from the server and save it in it’s cache and forward the response to the client.A screenshot of a computer

Description automatically generated

Text

Description automatically generated

As we can see here the response is saved in the cache.

A screenshot of a computer

Description automatically generated

Now if we tried to get that link again we will see that the proxy server looked first in his cache it is already stored it or not, if yes as we can see the proxy server will sent the cached response to the client.

Graphical user interface, text, application

Description automatically generated

Here if we tried to ender a website that is blocked by the proxy server

The proxy server will check first if this is blocked site or not, if yes he will refuse to access it to the client and will show him message says this URL is blocked

Graphical user interface, text

Description automatically generated

Here if we tried to enter an invalid URL the proxy server will fire an exception that handled by showing the client message says “Error 404 this page is not found”