Homework 2

To be released by Saturday October 25th

Bh2334@nyu.edu

Homework submissions typeset in LATEX are preferred (you might want to use https:// www.overleaf.com). If your name is Alan Turing, please upload a file called Turing.pdf. As PDF is the only accepted file format, make sure you do not submit a docx or jpg file. You are encouraged to insert scanned figures and illustrations. Scanned handwritten submissions will only be graded if neatly written and perfectly legible.
Submission is through Brightspace → Assignments. Please use email only as a last resort solution.

Exercise 1: (1+1+1+1 = 4 points)
True or false?

A. A user requests a Web page that consists of some text and three images. For this page, the client will send one request message and receive four response messages.
B. Two distinct Web pages (for example, www.mit.edu/research.html and www.mit.edu/students.html) can be sent over the same persistent connection.
C. With nonpersistent connections between browser and origin server, it is possible for a single TCP segment to carry two distinct HTTP request messages.
D. The Date: header in the HTTP response message indicates when the object in the response was last modified.

Exercise 2: (1+1+2+1+2+2+1+1+2 = 13 points)
1) Consider the following string of ASCII characters that were captured by Wireshark when the browser sent an HTTP GET message (i.e., this is the actual content of an HTTP GET message). The characters <cr><lf> are carriage return and line-feed characters (that is, the italized character string <cr> in the text below represents the single carriage-return character that was contained at that point in the HTTP header). Answer the following questions, indicating where in the HTTP GET message below you find the answer.

GET /cs453/index.html HTTP/1.1*<cr><lf>*Host: gai a.cs.umass.edu*<cr><lf>*User-Agent: Mozilla/5.0 (Windows;U; Windows NT 5.1; en-US; rv:1.7.2) Gec ko/20040804 Netscape/7.2 (ax) *<cr><lf>*Accept:ex t/xml, application/xml, application/xhtml+xml, text /html;q=0.9, text/plain;q=0.8,image/png,*/*;q=0.5 *<cr><lf>*Accept-Language: en-us,en;q=0.5<cr><lf>Accept- Encoding: zip,deflate*<cr><lf>*Accept-Charset: ISO -8859-1,utf-8;q=0.7,*;q=0.7*<cr><lf>*Keep-Alive: 300<cr> <lf>Connection:keep-alive*<cr><lf><cr><lf>*

A. What is the URL of the document requested by the browser?
B. What version of HTTP is the browser running?
C. Does the browser request a non-persistent or a persistent connection?
D. What is the IP address of the host on which the browser is running?
E. What type of browser initiates this message? Why is the browser type needed in an HTTP request message?

2) The text below shows the reply sent from the server in response to the HTTP GET message in the question above. Answer the following questions, indicating where in the message below you find the answer.

HTTP/1.1 200 OK*<cr><lf>*Date: Tue, 07 Mar 2008 12:39:45GMT*<cr><lf>*Server: Apache/2.0.52 (Fedora) *<cr><lf>*Last-Modified: Sat, 10 Dec2005 18:27:46 GMT*<cr><lf>*ETag: ”526c3-f22-a88a4c80”*<cr><lf>*Accept- Ranges: bytes*<cr><lf>*Content-Length: 3874*<cr><lf>* Keep-Alive: timeout=max=100*<cr><lf>*Connection: Keep-Alive*<cr><lf>*Content-Type: text/html; charset= ISO-8859-1*<cr><lf><cr><lf>*<!doctype html public ”- //w3c//dtd html 4.0transitional//en”>*<lf>*<html>*<lf>* <head>*<lf>* <meta http-equiv=”Content-Type” content=”text/html; charset=iso-8859-1”>*<lf>* <meta name=”GENERATOR” content=”Mozilla/4.79 [en] (Windows NT 5.0; U) Netscape]”>*<lf>* <title>CMPSCI 453 / 591 / NTU-ST550ASpring 2005 homepage</title>*<lf>*</head>*<lf>* <much more document text following here (not shown)>

A. Was the server able to successfully find the document or not? What time was the document reply provided?
B. When was the document last modified?
C. How many bytes are there in the document being returned?
D. What are the first 5 bytes of the document being returned? Did the server agree to a persistent connection?

Exercise 3: (2+3 = 5 points)
Details can be found on the RFC 5321 for SMTP.

A. What does MTA stand for?
B. Consider the following received spam e-mail (modified from a real spam e-mail). Assuming only the originator of this spam e-mail is malicious, and all other hosts are honest, identify the malicious host that has generated this spam e-mail.

```
From - Fri Nov 07 13:41:30 2008
Return-Path: <tennis5@pp33head.com>
Received: from barmail.cs.umass.edu (barmail.cs.umass.edu
[128.119.240.3]) by cs.umass.edu (8.13.1/8.12.6) for
<hg@cs.umass.edu>; Fri, 7 Nov 2008 13:27:10 -0500
Received: from asusus-4b96 (localhost [127.0.0.1]) by
barmail.cs.umass.edu (Spam Firewall) for <hg@cs.umass.edu>;
Fri, 7 Nov 2008 13:27:07 -0500 (EST) Received: from asusus-
4b96 ([58.88.21.177]) by barmail.cs.umass.edu
for <hg@cs.umass.edu>; Fri, 07 Nov 2008 13:27:07 -0500
(EST) Received: from [58.88.21.177] by inbnd55.exchangeddd.
com; Sat, 8 Nov 2008 01:27:07 +0700
From: "Jonny" <tennis5@pp33head.com>
To: <hg@cs.umass.edu>
Subject: How to secure your savings
```

Exercise 4: (3+2 = 5 points)
In this problem, we use the useful dig tool available on Unix and Linux hosts to explore the hierarchy of DNS servers. Recall that, a DNS server in the DNS hierarchy delegates a DNS query to a DNS server lower in the hierarchy, by sending back to the DNS client the name of that lower-level DNS server. First read the man page for dig, and then answer the following questions.

A. Starting with a root DNS server (from one of the root servers [a-m]. root-servers.net), initiate a sequence of queries for the IP address for your department's Web server by using dig. Show the list of the names of DNS servers in the delegation chain in answering your query.
B. Repeat part (A) for several popular Web sites, such as google.com, yahoo .com, or amazon.com.

Exercise 5: (5+5+5+5 = 20 points):
Create a small application for a chat-bot with Python. The server must reply to some few questions (feel free about the questions) to a client (without a Graphical User Interface (GUI)). Provide two versions of the application: the first with a non-reliable transport service and the second with a reliable one.

- Upload code files on Brightspace. Your code must be well commented especially regarding the methods you use.
- Provide your code also on the PDF with instructions on how to run the code files provided as well as a screenshot of the execution (show questions – replies of the chatbot).