SAINT MARY'S UNIVERSITY SINCE 1802 | FACULTY OF SCIENCE

CSCI 3421.1 Fall, 2018 – Assignment 4

Data Communications & Network

Due Date: Friday, November 2, 2018
Closing Date: Sunday, November 4, 2018

Deadline Policy

There will be a due date and a closing date for each assignment

- If you submit your work after the due date (but before the closing date), a *late submission* is applied to your overall assignments submission.
- You are allowed to use a *late submission* for up to 3 assignments throughout the semester without being penalized. After that, your assignment grade will be penalized by 10%.
- You will not be able to submit or receive credit for your work after the closing date.

Submission

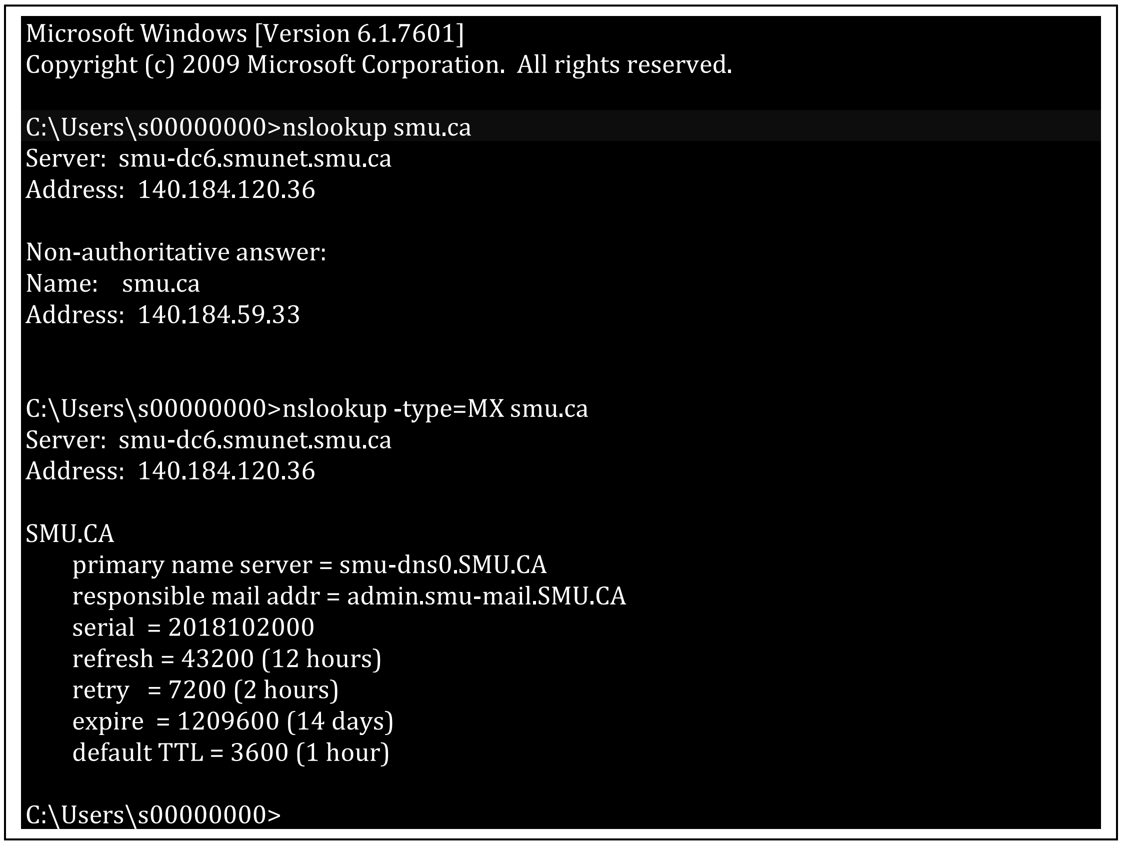
Once you are done with your assignment, make sure you do the following before the assignment due date:

- Prepare a report containing the answers to the assignment questions and/or self-evaluation.
- Include your Name and A#.
- Create one PDF file for your report.
- Combine your source code and PDF report into one zip file.
- Name your file following this convention: CSCI3421-Assignment4-*StudentID*, where *StudentID* is your Banner ID number starting with A.
- Submit your Zip file via Brightspace at https://smu.brightspace.com before the deadline.
- Your programs will be compiled and tested on the university machines.

Assignment Description

Question 1: [2 points] **DNS** and **nslookup (command line)**
In its most basic operation, *nslookup* allows querying any specified DNS server for a DNS record. The queried DNS server can be a root DNS server, a top-level-domain DNS server, an authoritative DNS server, or an intermediate DNS. To perform this task, *nslookup* sends a DNS query to the specified DNS server, receives a DNS reply from that same DNS server, and displays the result.

Below is the result of two independent nslookup commands. When the DNS server name is not provided, the default DNS server is used. The first *nslookup* is requesting the IP address for the host "smu.ca". The second nslookup is requesting the IP address for the mail server associated with the host "smu.ca".



```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\s00000000>nslookup smu.ca
Server:  smu-dc6.smunet.smu.ca
Address:  140.184.120.36

Non-authoritative answer:
Name:    smu.ca
Address:  140.184.59.33


C:\Users\s00000000>nslookup -type=MX smu.ca
Server:  smu-dc6.smunet.smu.ca
Address:  140.184.120.36

SMU.CA
        primary name server = smu-dns0.SMU.CA
        responsible mail addr = admin.smu-mail.SMU.CA
        serial  = 2018102000
        refresh = 43200 (12 hours)
        retry   = 7200 (2 hours)
        expire  = 1209600 (14 days)
        default TTL = 3600 (1 hour)

C:\Users\s00000000>
```

<u>Your turn….</u>

1) Run nslookup so that Google public DNS server is queried for the mail servers for Gmail!.
2) Run nslookup to obtain the IP address of the virtual server netsec.cs.smu.ca

Question 2: [4 points] **DNS** and **ipconfig**
Ipconfig (for Windows) and *ifconfig* (for Linus/Unix) are very useful utilities in your host, especially for debugging network issues. Find out what *ipconfig*/*ifconfig* is about and show some at least two different examples of how you can use it.

Question 3: [6 points] **Wireshark** and **UDP**
For this exercise, you will use a packet sniffer (e.g., Wireshark) to capture and examine packets. If you are not familiar with how to use Wireshark, consult the Wireshark_intro (textbook supplement) available on the course website.

Start capturing packets in Wireshark and then do something that will cause your host to send and receive several UDP packets. (One way to do this would be to use *nslookup*).

After stopping packet capture, set your packet filter so that Wireshark only displays the UDP packets sent and received at your host and then answer the following questions:

1) Select one UDP packet. From this packet, determine how many fields there are in the UDP header. Name these fields.
2) By consulting the displayed information in Wireshark's packet content field for this packet, determine the length (in bytes) of each of the UDP header fields.
3) What is the largest possible source port number?
4) What is the protocol number for UDP? Give your answer in both hexadecimal and decimal notation. (To answer this question, you'll need to look into the IP header.)
5) Examine a pair of UDP packets in which the first packet is sent by your host and the second packet is a reply to the first packet. Describe the relationship between the port numbers in the two packet

Guidelines
Due to privacy policy, you will not capture packets on the university live network. If you wish to answer this question while on campus, then you will need to log into our cs virtual server: netsec.cs.smu.ca.

- You will need to contact Mr. Nikita Neveditsin at Nikita.Neveditsin@smu.ca in order to get your login information for that server.
- You can use SSH or PuTTY to remotely login to the server.
- You can run tshark, a command line version of Wireshark to capture packets.
- Here is how you can use tshark:
  - tshark –i ens3 –a duration:5 → will capture packets for 5 seconds
  - tshark –i ens3 –c 10 → will capture 10 packets and stop
  - tshark –i ens3 –w test.pcap → will capture packets and write the output onto the file test.pcap
  - tshark udp –a duration:5 → will capture UDP packets for 5 seconds
  - tshark –r test.pcap → will read the content of the file and displayed it
  - capinfos test.pcap → will display detailed information about the pcap file

References:
https://putty.org/
https://www.wireshark.org/docs/man-pages/tshark.html

Question 4: [10 points] **Programming with Sockets - Mail Client**
The purpose of this exercise is to help you acquire a better understanding of SMTP protocol and gain more experience in implementing a standard protocol using a programming language of your choice.

Your task is to develop a simple mail client that sends a single email (text based) to any recipient. Your client will need to connect to a mail server, dialogue with the mail server using the SMTP protocol, and send an email message to the mail server.

NOTES

- In order to limit spam, some mail servers do not accept TCP connection from arbitrary sources. You may want to try connecting both to your university mail server and to a popular Webmail server. You may also try making your connection both from your home and from your university campus.

- In some cases, the receiving mail server might classify your e-mail as junk. Make sure you check the junk/spam folder when you look for the e-mail sent from your client.

- Mail servers like Google mail (address: smtp.gmail.com, port: 587) requires your client to add a Transport Layer Security (TLS) or Secure Sockets Layer (SSL) for authentication and security reasons, before you send MAIL FROM command. Add TLS/SSL commands to your existing ones and implement your client using Google mail server at above address and port.

- If Python is your language of choice, do not use **smtplib** because it hides the details of SMTP and socket programming.

BONUS [3 points]: Modify your client such that it can send emails with both text and images.

Your job is to:
Task 1: (5 points) Write a `mailClient`
Task 2: (**2 points**) Test your mail client using various mail servers.
Task 3: (**2 points**) Capture the packets sent by your mail client using wireshark (Question3)
Task 4: (**1 point**) Explain the packets captured

What to hand in? You will submit the complete `mailClient` code along with the screen shots of your program runs, verifying that your client is able to connect to the mail server and send/receive the responses. You will also submit the .pcap file from your tshark capture.

Evaluation Criteria

- **Correctness:** Program is free of syntax and logical errors. Program must meet the requirements specified in the assignment description. Proper handling of exceptions, error conditions, and special cases as well as providing correct results/output**.**
- **Testing:** test cases provided take into consideration all cases including error conditions and special cases. Test your mail client using various mail servers.
- **Style:** code is general purpose and well organized
- **Documentation:** Your program is properly documented; at the beginning of each class/function, there is a brief description about the class/function, its main characteristics, main functionality and proper usage.
- **Efficiency:** program should use the most appropriate solution/method
- **Interface:** program must be simple to use. Your program uses simple and clear messages when prompting the user to enter data. Your program notifies the user with a closing message when the program is terminated.

Self-evaluation

Please answer the following questions:

1. **[1 points]** Were you able to complete this assignment? What grade are you expecting? Justify.
2. **[1 points]** Describe 2-3 challenges you faced while completing this assignment. How did you tackle those challenges?
3. **[1 points]** Provide a break down for the activities/milestones for this assignment. Give an estimate of hours spent on each activity. Try to be honest!

Date	Activities	Hours	Outcome

Evaluation Scheme

Student Name:	
Student ID:	
Total Points	**/25**
Programming Exercises	**/22**
Self-evaluation questions	**/3**

Exercise	1	2	3	4	Bonus
Points	2	4	6	10	3
Score					