Due before:	**11:59 PM (before Midnight) on Wednesday April 29, 2020**

CSE278: Introduction to Systems Programming (Systems I)

Homework #8

Due: Wednesday April 29, 2020 before 11:59 PM

Email-based help Cutoff: 5:00 PM on Tue, April 28, 2020

Maximum Points: 50

Submission Instructions

This part of the homework assignment must be turned-in electronically via Canvas. Ensure you name this document *HW8_MUID*.docx, where *MUid* is your Miami University unique ID. (Example: HW8_ahmede.docx)

Copy pasting from online resources is **Plagiarism**. Instead you should read, understand, and use your own words to respond to questions.

Submission Instructions:
Once you have completed answering the questions save this document as a PDF file (**don't just rename the document; that is not the correct way to save as PDF**) and upload it to Canvas.

General Note: Upload each file associated with homework (or lab exercises) individually to Canvas. Do not upload archive file formats such as zip/tar/gz/7zip/rar etc.

Objective

The objective of this homework is to review basic concepts of:

- Basics of Pointers and Vulnerability
- Basics of Systems Security
- Experiment with phishing via SMTP protocol

Name: **Ben Hilger**

Required reading

- Lecture Slides Pointers
- ClassNotes Pointers
- Lecture Slides Security
- ClassNotes Security

1. What does it mean to dereference a pointer?

When dereferencing a pointer it means to get the value that is stored at the memory location that the pointer is referencing.

2. How are pointers different than references?

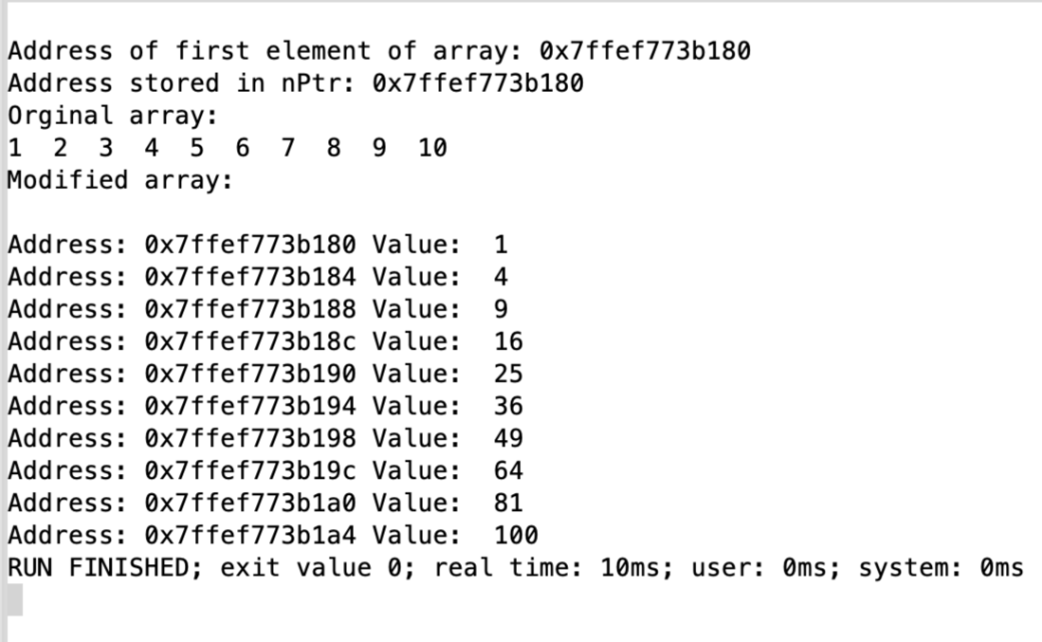
Pointers are different from references because pointers are a memory address for an object, and therefore can also change to reference new memory addresses when needed. Whereas references are more of an alias of a specific object and can’t be changed once initialized. This means that a reference can only change the value of the object it is representing, whereas pointers can use arithmetic to change memory addresses/the object they are referencing or the value of the object they are currently pointing at.

3. What does this program do?

```
#include <iostream>
using namespace std;

int main()
{
    int numbers[10] {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
    int* nPtr;

    nPtr = &numbers[0];

    cout << "\nAddress of first element of array: " << &numbers[0];
    cout << "\nAddress stored in nPtr: " << nPtr;

    cout << "\nOrginal array: \n";
    for (size_t i {0}; i < 10; i++)
        cout << numbers[i] << "  ";


    for (size_t j {0}; j < 10; j++) {
         *(nPtr + j) = *(nPtr + j) * *(nPtr + j);
     }
    cout << "\nModified array: \n";
    for (size_t k {0}; k < 10; k++) {
        cout << "\nAddress: " << (nPtr + k) << "\tValue:  " << *(nPtr + k);
    }
    return 0;
```

```
            }
```

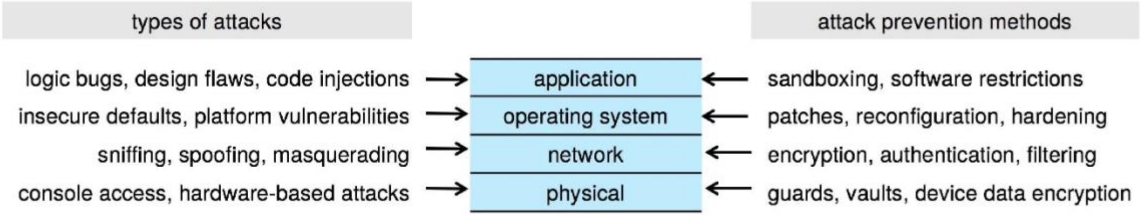
Attach a screen shot of sample run



```
Address of first element of array: 0x7ffef773b180
Address stored in nPtr: 0x7ffef773b180
Orginal array:
1  2  3  4  5  6  7  8  9  10
Modified array:

Address: 0x7ffef773b180 Value:  1
Address: 0x7ffef773b184 Value:  4
Address: 0x7ffef773b188 Value:  9
Address: 0x7ffef773b18c Value:  16
Address: 0x7ffef773b190 Value:  25
Address: 0x7ffef773b194 Value:  36
Address: 0x7ffef773b198 Value:  49
Address: 0x7ffef773b19c Value:  64
Address: 0x7ffef773b1a0 Value:  81
Address: 0x7ffef773b1a4 Value:  100
RUN FINISHED; exit value 0; real time: 10ms; user: 0ms; system: 0ms
```

Security

4. Four-layered Model of Security


types of attacks
attack prevention methods
logic bugs, design flaws, code injections
application
sandboxing, software restrictions
insecure defaults, platform vulnerabilities
operating system
patches, reconfiguration, hardening
sniffing, spoofing, masquerading
network
encryption, authentication, filtering
console access, hardware-based attacks
physical
guards, vaults, device data encryption

Explain the above figure in view of security.

This diagram shows the four different layers of security for a computer, in order of most visible at the top to the least visible at the bottom. Each layer has a different set of vulnerabilities and prevention methods to best protect it from attacks as outlined in the diagram on the left (attacks) and the right (prevention). The four layers are described below:

This figure shows the different layers of security in a computer. The first layer is the application layer and is the top layer of the computer infrastructure and therefore is seen by the user. This is where the software lies, and there are certain attacks such as logic bugs and flaws that can infiltrate this layer. There are also ways of preventing these attacks such as sandboxing as shown in the diagram.

The next layer is the operating system and is right below the application layer. This contains OSs such as MacOS, Windows and Linux. These are massive applications and therefore are subject to attacks such as insecure defaults and platform vulnerabilities. There are also ways to prevent attacks and infiltrations at this level such as patches and reconfiguration as shown in the diagram.

The next layer is the network layer, which is where packets are exchanged. On the left side of the network layer in the diagram is the common attacks such as sniffling, spoofing, and masquerading attacks. On the right of the network layer is the common methods to help prevent those attacks such as encryption, authentication and filtering.

The final layer is the physical layer, which is the lowest layer and hardest to see by the user. As shown in the diagram, common attacks against this layer include console access and hardware-based attacks and common prevention methods include things such as vaults and guards.

So, this diagram outlines the four main layers of a computer that are vulnerable to attack and shows each layer's means of prevention. So, by breaking it down into these four layers as the diagram has done, it's easier to see where and how to best protect each layer, which in turn protects the whole system overall.

Step 5: Sending email via SMTP (to yourself)

Background: Similar to HTTP sending emails is accomplished using a simple text protocol called Simple Mail Transfer Protocol (SMTP). Similar to all the protocols, SMTP is a text protocol that requires specific commands with suitable format and data.

Exercise: This exercise requires you to send yourself an email via SMTP using the following procedure:

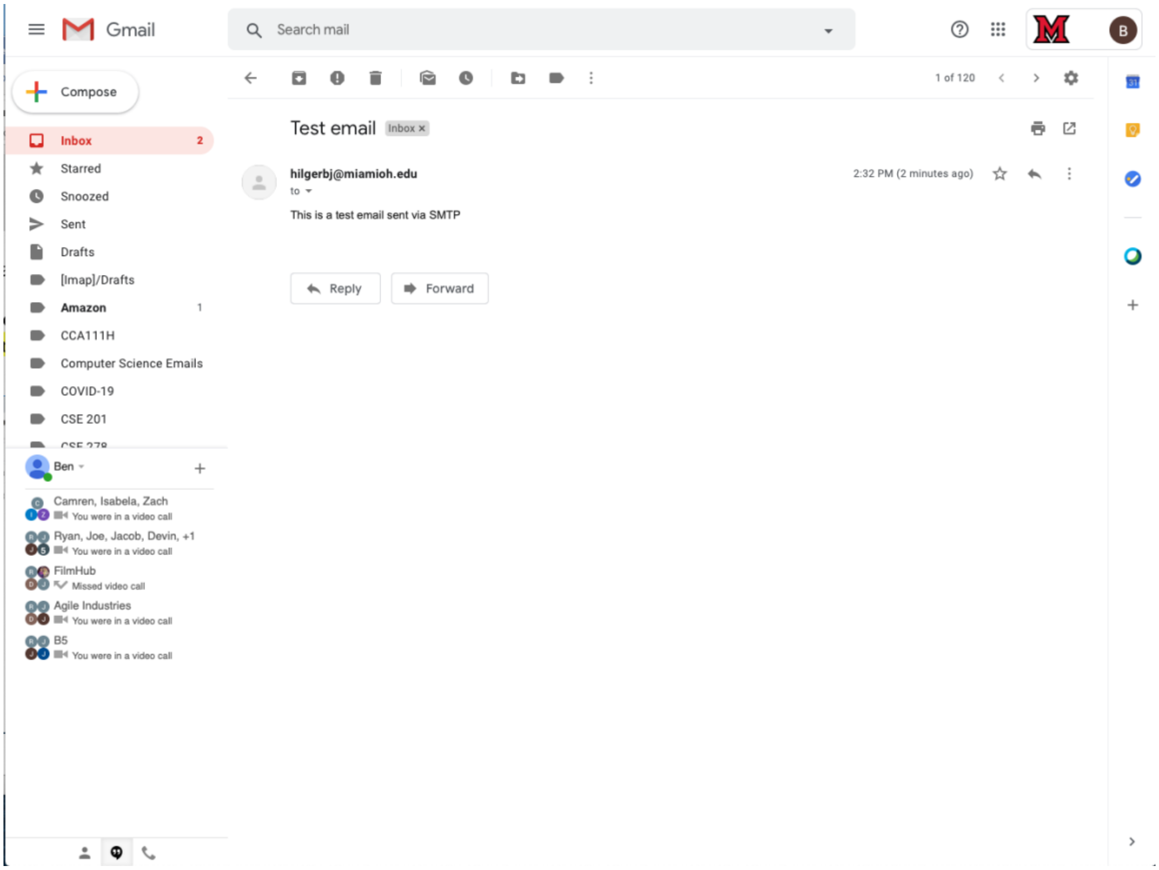
1. From a Terminal `ssh` into `os1.csi.miamioh.edu`.
2. Initiate a session with Miami University's mail forwarding server (on port 25) on `ceclnx01` using `telnet` as shown below:

```
$ telnet mailfwd.miamioh.edu 25
```

3. Using the commands, send yourself an email (at your Miami University account) using SMTP protocol by typing the following information show in **red**:

```
HELO os1.csi.miamioh.edu
250 mualmaip14.mcs.miamioh.edu
MAIL FROM: MUID@miamioh.edu
250 2.1.0 Ok
RCPT TO: <MUID@miamioh.edu>
250 2.1.5 Ok
DATA
354 End data with <CR><LF>.<CR><LF>
From: YourName <MUID @miamioh.edu>
Subject: Test email
This is a test email sent via SMTP.
.
250 2.0.0 Ok: queued as E40CA2BE7C
QUIT
```

4. The above SMTP commands will send out an email to your Miami University email account. You should receive the email at your Miami University account. Make a screenshot of the email you sent (as seen in `gmail` mail reader) and past it in the space below replacing the example screenshot shown below:



Test email

hilgerbj@miamioh.edu

to

2:32 PM (2 minutes ago)

This is a test email sent via SMTP

STEP 6: Phishing via email

Background: Phishing is the attempt to obtain information such as usernames, passwords, and credit card details (and sometimes, indirectly, money), by masquerading as a trustworthy entity. Phishing via email is often accomplished by masquerading email from a trustworthy authority. SMTP inherently does not have an effective approach for validating sender's email address -- but there are a few precautions that mail server provide (but they are not 100%)

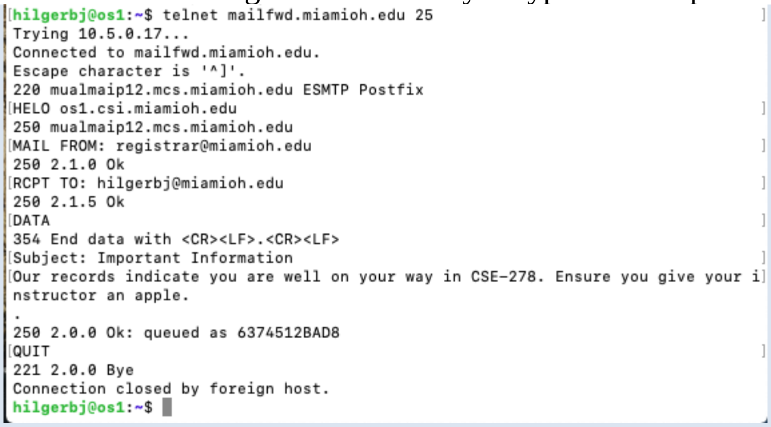
Exercise: In this exercise you are expected to send an email to your neighbor in the lab using SMTP protocol. Most of the procedure is similar to the previous part:

1. From a Terminal `ssh` into `os1.csi.miamioh.edu`.
2. Ask your neighbor's email address and inform them you are planning to send them a phishing email.
3. Initiate a session with Miami University's mail forwarding server (on port 25) using `telnet` as shown below:

   ```
   $ telnet mailfwd.miamioh.edu 25
   ```
4. In your SMTP commands change the `MAIL FROM` option to:

   ```
   MAIL FROM: <registrar@miamioh.edu>
   ```
5. In your SMTP commands change the `RCPT TO` option to your neighbor's email address.
6. Ensure the subject and email contents is set to:
   a. `Subject: Important information`
   b. Message -- `Our records indicate you are well on your way in CSE-278. Ensure you give your instructor an apple.`

7. Once your email has been accepted for delivery, make a screenshot of the terminal showing the commands you typed in the space below:



```
hilgerbj@os1:~$ telnet mailfwd.miamioh.edu 25
Trying 10.5.0.17...
Connected to mailfwd.miamioh.edu.
Escape character is '^]'.
220 mualmaip12.mcs.miamioh.edu ESMTP Postfix
HELO os1.csi.miamioh.edu
250 mualmaip12.mcs.miamioh.edu
MAIL FROM: registrar@miamioh.edu
250 2.1.0 Ok
RCPT TO: hilgerbj@miamioh.edu
250 2.1.5 Ok
DATA
354 End data with <CR><LF>.<CR><LF>
Subject: Important Information
Our records indicate you are well on your way in CSE-278. Ensure you give your i
nstructor an apple.
.
250 2.0.0 Ok: queued as 6374512BAD8
QUIT
221 2.0.0 Bye
Connection closed by foreign host.
hilgerbj@os1:~$ 
```

Due before:	**11:59 PM (before Midnight) on Wednesday April 29, 2020**

Submission

- No late assignments will be accepted!
- This work is to be done individually
- The submission file will be saved with the name ***HW8_yourMUID.pdf***
- Assignment is due before Midnight Wednesday April 29, 2020.
- On or before the due time, drop the *electronic copy* of your work in the *canvas*
- Don't forget to Turn in the file!   HW8_yourMUID.pdf