Luke Canfield ECN Login: lcanfiel Due Date: 4/4/2024

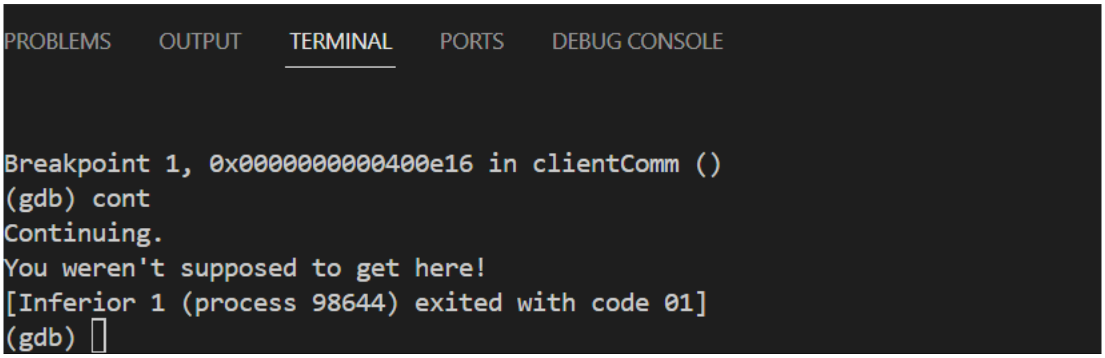
HW10 ECE 40400

<u>Special Buffer Overflow String</u>

AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA\x18\x0e\x40\x00

<u>Determining the Special String</u>

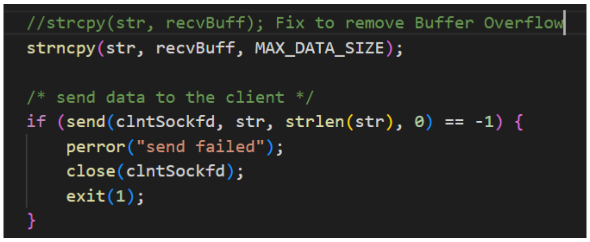
I followed the instructions in Lecture 21 starting at page 43 to determine the string and I will briefly go through the steps I used as the process for this homework is slightly different than the example given. I compiled the server code and then opened the executable in gdb. Then I disassembled the secretFunction to locate the first memory address to enter into the object code for this function. When we overwrite the return address of the clientComm stack frame we want it to be overwritten with this memory address to enter the secret function. We only needed to use the last four bytes of the address. Then I disassembled the clientComm function and found the address of the “leaveq” assembly function so that I could set a breakpoint at that specific address. Then I ran the code with the port being 9000. I opened another terminal and compiled the client.c file and ran the executable as ./client 127.0.0.1. Then I was prompted to to type in a string and I entered 10 A's. Going back to the server terminal it had stopped at the specified breakpoint. Then I used print /x *((unsigned *) $rbp + 2) to see the return address of the stack frame and used x /48b $rsp to see the first 48 bytes of the stack frame being pointed to by the stack pointer. I could see the hex values of the 10 A’s and looking at the return address of the stack frame I was able to count how many A’s would need to be sent to where I could append the reverse of the last four bytes of the first memory address in the secretFunction to overwrite the return address of the stack frame. Then I went back to the client terminal and inserted the string. Going back to the server terminal I could see the secretFunction had been entered as the print statement had been called from the function as seen below.



```
Breakpoint 1, 0x0000000000400e16 in clientComm ()
(gdb) cont
Continuing.
You weren't supposed to get here!
[Inferior 1 (process 98644) exited with code 01]
(gdb) 
```

<u>Fix to server.c</u>

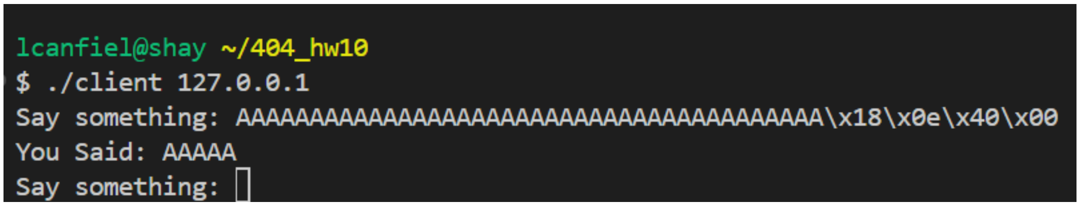
I simply used strncpy instead of strcpy so that I could specify at most n characters to be copied where n is the MAX_DATA_SIZE. This ignores all other characters after the MAX_DATA_SIZEth character.



```
//strcpy(str, recvBuff); Fix to remove Buffer Overflow
strncpy(str, recvBuff, MAX_DATA_SIZE);

/* send data to the client */
if (send(clntSockfd, str, strlen(str), 0) == -1) {
    perror("send failed");
    close(clntSockfd);
    exit(1);
}
```

The image below also shows that even though the special buffer overflow string was sent the first five As are all that was copied so the buffer overflow attack doesn't occur anymore. Before hand the "You Said:" contained 40 As with the @ symbol afterwards.



```
lcanfiel@shay ~/404_hw10
$ ./client 127.0.0.1
Say something: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA\x18\x0e\x40\x00
You Said: AAAAA
Say something: 
```

Page worth of Log Contents

```
New message log:1

procmail: Couldn't determine implicit lockfile from "/usr/sbin/sendmail"

From lcanfiel@purdue.edu  Thu Apr  4 10:20:14 2024

 Subject: Email Test

  Folder: /usr/sbin/sendmail -oi lcanfiel@purdue.edu                    6155

New message log: 2

From bounces+8143873-a96e-ece404e1=ecn.purdue.edu@u8143873.wl124.sendgrid.net  Thu Apr  4 10:38:40 2024

 Subject: =?UTF-8?B?WW914oCZcmU=?= now signed up to Down to Earth

  Folder: spamFolder                                                68759

New message log: 3

From bounces+8143873-a96e-ece404e1=ecn.purdue.edu@u8143873.wl124.sendgrid.net  Thu Apr  4 10:38:41 2024

 Subject: =?UTF-8?B?WW914oCZcmU=?= now signed up to First Thing

  Folder: spamFolder                                                69088

New message log: 4

From bounces+8143873-a96e-ece404e1=ecn.purdue.edu@u8143873.wl124.sendgrid.net  Thu Apr  4 10:38:41 2024

 Subject: Welcome to Well Actually

  Folder: spamFolder                                                68839

From bounces+8143873-a96e-ece404e1=ecn.purdue.edu@u8143873.wl124.sendgrid.net  Thu Apr  4 10:38:40 2024

 Subject: =?UTF-8?B?WW914oCZcmU=?= now signed up to Margaret Sullivan's
```

```
  Folder: spamFolder                              69367

From bounces+8143873-a96e-ece404e1=ecn.purdue.edu@u8143873.wl124.sendgrid.net  Thu Apr  4
10:38:40 2024

 Subject: =?UTF-8?B?WW914oCZcmU=?= now signed up to Robert Reich's newsletter

  Folder: spamFolder                              69098

From bounces+8143873-a96e-ece404e1=ecn.purdue.edu@u8143873.wl124.sendgrid.net  Thu Apr  4
10:38:40 2024

 Subject: =?UTF-8?B?WW914oCZcmU=?= now signed up to Soccer with Jonathan Wilson

  Folder: spamFolder                              69552

From bounces+8143873-a96e-ece404e1=ecn.purdue.edu@u8143873.wl124.sendgrid.net  Thu Apr  4
10:38:40 2024

 Subject: =?UTF-7?B?WW914oCZcmU=?= now signed up to Trump on Trial

  Folder: spamFolder                              68795

New message log: 5

From bounces+8143873-a96e-ece404e1=ecn.purdue.edu@u8143873.wl124.sendgrid.net  Thu Apr  4
10:39:56 2024

 Subject: Welcome to Reclaim Your Brain

  Folder: spamFolder                              74194
```