Ethan James
Epjames
33378430

HW10

Buffer Overflow Attack Problem:

```
RECEIVED: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA@RECEIVED BYTES: 43

You weren't supposed to get here!
[Inferior 1 (process 169692) exited with code 01]
(gdb)
```

```
Say something: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA\x18\x0e\x40\x00
You Said: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA@
```

I decided to use 40 “A” characters and the reverse endianness of the start address of the secret function. The way I determined this was to print the registers. I determined that I needed to fill the A’s up until the return address of the function my client comm. This meant I needed 40 A’s. Once I had that figure out, I figured out that I now wanted to enter the secret function. To enter the secret function, I needed to replace the return address of clientComm with the starting address of secretFunction. I did this by using reverse endianness of the start address and placed this onto the return address of client Comm therefore entering the secretFunction.

```
//*****************************************************************************************************************************************
//Add code for avoiding buffer overflow
if(numBytes > MAX_DATA_SIZE + 1) //Added to avoid overflowing since numbytes would take up too much space, suceptible to overflow
{
    fprintf(stderr,"ERROR, Buffer Overflow\n");
    exit(1); //Terminate the program if overflow exists that way nothing can get changed and no attack possible
}
//*****************************************************************************************************************************************
strncpy(str, recvBuff,MAX_DATA_SIZE); //May need to take this out, this is what will cause the overflow
```

Here is my fix to the problem. I made two changes. I said if numBytes is greater than the max data size, then I would want to alert the system to a Buffer Flow and it would terminate without allowing me to gain access to things I should not be able to touch. I also altered the string copy line. I changed it to strncpy. Strncpy copies a specified number of characters from the source string to the destination, ensuring that the destination buffer won't overflow and optionally adding a null terminator if necessary. This allows me to make sure that the buffer overflow attack won’t occur. It won’t let the string get big enough to cause the attack.

Spam Overflow Attack Problem:

```
New message log:
4
From bounce@beauty.sephora.com  Tue Apr  2 15:40:08 2024
 Subject:
  Folder: spamFolder                                                  45321


New message log:
5
From bounce-19_HTML-178147532-25053-534000654-49105@bounce.email.jcpenney.com  Tue Apr  2 15:40:33 2024
 Subject: Here's a little welcome gift just for you!
  Folder: spamFolder                                                  56782


New message log:
6
From bounce-39_HTML-155321549-214403-514005125-316@bounce.e.lululemon.com  Tue Apr  2 15:41:25 2024
 Subject: Welcome: 15% off
  Folder: spamFolder                                                  42435


New message log:
7
From bounce-290_HTML-711022075-2335539-10990156-7642423@bounce.e.fanaticsretailgroup.com  Tue Apr  2 15:43:09 2024
 Subject: Welcome To Fanatics
  Folder: spamFolder                                                  56047


New message log:
8
From bounce-26_HTML-131886304-752597-7232476-544@bounce.e.ralphlauren.com  Tue Apr  2 15:45:46 2024
 Subject: Welcome to the World of Ralph Lauren
  Folder: spamFolder                                                  77800


New message log:
9
From bounce-tj6-ZH_marshalls_NNTAN04022024c1193379b0-h-2354bbe8c3=2@eml.marshalls.com  Tue Apr  2 15:46:55 2024
 Subject: You're officially in!
  Folder: spamFolder                                                  99970
```