Terrence Randall
Randall7
4/9/20

Specially crafted string for overflow attack:

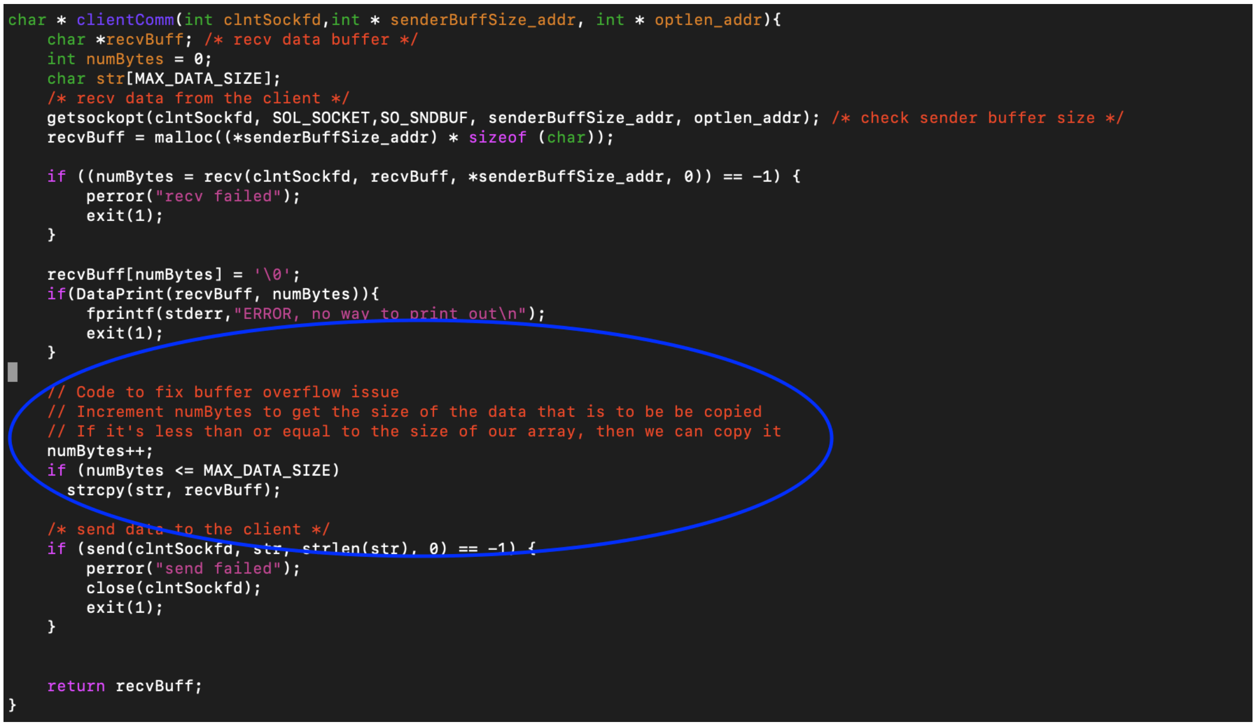
After obtaining an entrance address into the secret function, which I found to be 0x00 0x00 0x00 0x0f 0x3b, I attempted to send in every combination from A\x3b\x0f to {A x 50}\x3b\x0f to get the secret function to execute.

But I wasn’t able to get the function to execute.

Rationale behind decision:

After a failure to analyze the system stack, an exhaustive approach seemed like the best option.

Modified Server Code:



```c
char * clientComm(int clntSockfd,int * senderBuffSize_addr, int * optlen_addr){
    char *recvBuff; /* recv data buffer */
    int numBytes = 0;
    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);
    }

    // Code to fix buffer overflow issue
    // Increment numBytes to get the size of the data that is to be be copied
    // If it's less than or equal to the size of our array, then we can copy it
    numBytes++;
    if (numBytes <= MAX_DATA_SIZE)
      strcpy(str, recvBuff);

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


    return recvBuff;
}
```

Notes on Changes made:

The initial vulnerability was present because there was no check on the size of the data being taken in as an input. And if the size was bigger than the area allocated on the stack, when the input data is copied over to the stack variable it will overwrite data beyond its bound. This means that with the right input string an attacker can overwrite the return address of the function to be the entry point into the “secretFunction”.

To fix this vulnerability, I first added an increment to “numBytes” so that it would then be the size of the array that was attempting to be copied. After the increment, I checked that

Terrence Randall
Randall7
4/9/20

the value of numBytes was less than or equal to the size of the array. If it is then the program may execute the string copy, otherwise the array simply won’t change.