Buffer Overflow String

The specially-crafted buffer overflow string I developed for this assignment is:
AAAAAAAAAAAAAAAAAAAAAAAA\xe3\x16\x00\x00\x00\x00\x00\x00

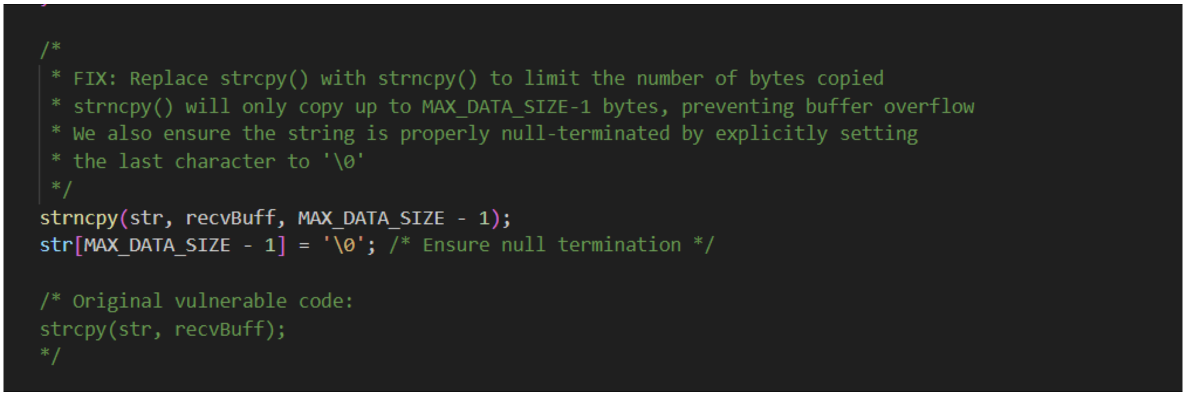
Explanation of How I Determined the Special String

To determine the buffer overflow string, I first identified the vulnerability in the server.c program. By examining the code, I found that the clientComm() function uses strcpy() to copy data from the recvBuff to a fixed-size buffer str[MAX_DATA_SIZE] without any bounds checking. Since MAX_DATA_SIZE is only 5 bytes, any input larger than this would overflow the buffer. Using GDB, I identified the memory address of secretFunction() as 0x00000000000016e3. Through stack analysis, I determined that approximately 24 bytes of padding were needed to reach the return address on the stack. I crafted an exploit string consisting of 24 'A' characters to fill the buffer and reach the return address, followed by the address of secretFunction in little-endian format (bytes reversed). When this string is processed by the vulnerable strcpy() function, it overflows the buffer and overwrites the return address, causing the program to jump to secretFunction() instead of returning to its intended location.

Explanation of Fixes to server.c

The vulnerability in server.c stems from using strcpy() without any bounds checking. This allows an attacker to write beyond the bounds of the buffer and potentially overwrite critical stack data including the return address. To fix this vulnerability, I replaced the unsafe strcpy() function with strncpy(), which limits the number of bytes copied to a specified maximum. Additionally, I ensured proper null termination of the string by explicitly setting the last character of the buffer to '\0'. These changes prevent buffer overflow by ensuring that no more data is copied than the buffer can hold, regardless of the input size. The modified code now properly handles large inputs without compromising memory safety, effectively mitigating the buffer overflow vulnerability.

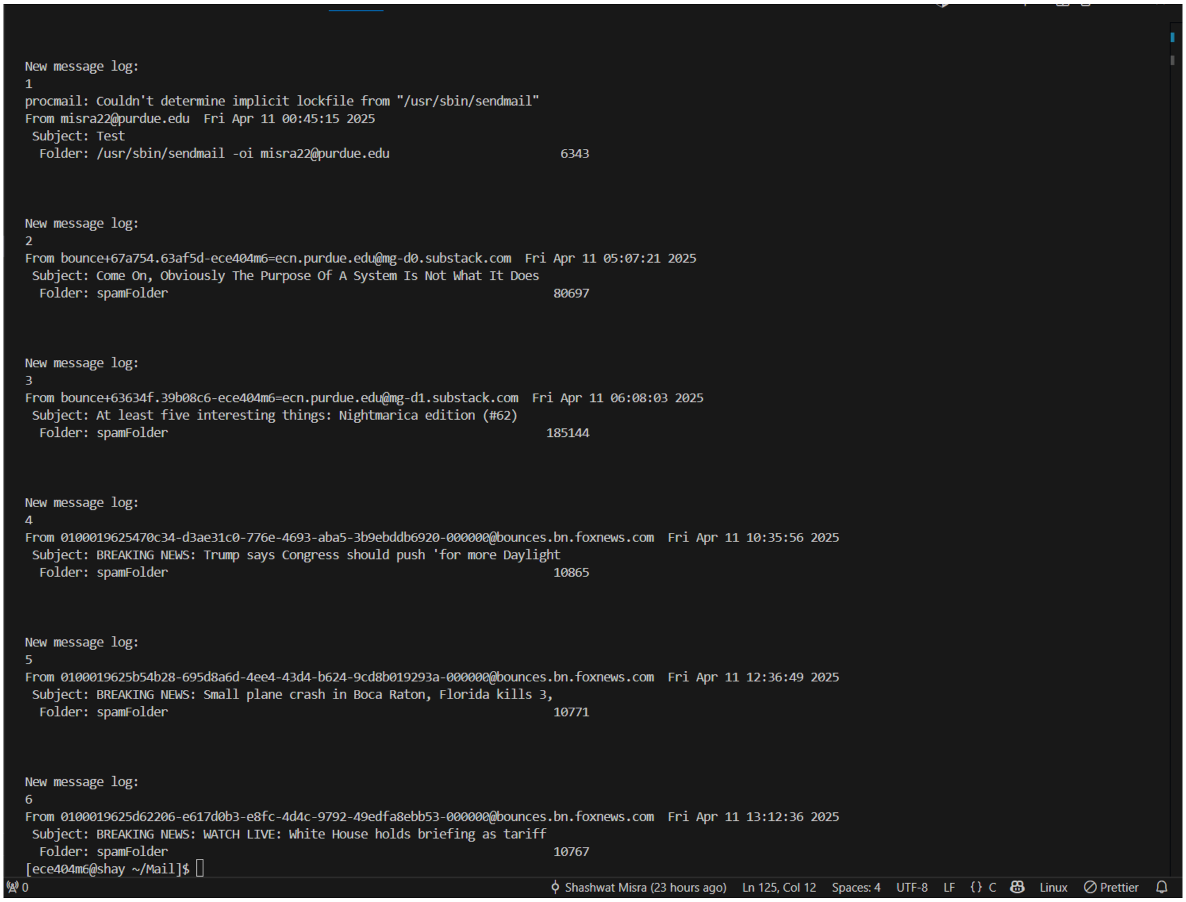
ORIGINAL VULNERABLE CODE AND FIX IMAGE BELOW



```c
/*
 * FIX: Replace strcpy() with strncpy() to limit the number of bytes copied
 * strncpy() will only copy up to MAX_DATA_SIZE-1 bytes, preventing buffer overflow
 * We also ensure the string is properly null-terminated by explicitly setting
 * the last character to '\0'
 */
strncpy(str, recvBuff, MAX_DATA_SIZE - 1);
str[MAX_DATA_SIZE - 1] = '\0'; /* Ensure null termination */

/* Original vulnerable code:
strcpy(str, recvBuff);
*/
```

Mail Directory Logfile Contents



```
New message log:
1
procmail: Couldn't determine implicit lockfile from "/usr/sbin/sendmail"
From misra22@purdue.edu  Fri Apr 11 00:45:15 2025
 Subject: Test
  Folder: /usr/sbin/sendmail -oi misra22@purdue.edu                        6343



New message log:
2
From bounce+67a754.63af5d-ece404m6=ecn.purdue.edu@mg-d0.substack.com  Fri Apr 11 05:07:21 2025
 Subject: Come On, Obviously The Purpose Of A System Is Not What It Does
  Folder: spamFolder                                                      80697



New message log:
3
From bounce+63634f.39b08c6-ece404m6=ecn.purdue.edu@mg-d1.substack.com  Fri Apr 11 06:08:03 2025
 Subject: At least five interesting things: Nightmarica edition (#62)
  Folder: spamFolder                                                     185144



New message log:
4
From 0100019625470c34-d3ae31c0-776e-4693-aba5-3b9ebddb6920-000000@bounces.bn.foxnews.com  Fri Apr 11 10:35:56 2025
 Subject: BREAKING NEWS: Trump says Congress should push 'for more Daylight
  Folder: spamFolder                                                      10865



New message log:
5
From 0100019625b54b28-695d8a6d-4ee4-43d4-b624-9cd8b019293a-000000@bounces.bn.foxnews.com  Fri Apr 11 12:36:49 2025
 Subject: BREAKING NEWS: Small plane crash in Boca Raton, Florida kills 3,
  Folder: spamFolder                                                      10771



New message log:
6
From 0100019625d62206-e617d0b3-e8fc-4d4c-9792-49edfa8ebb53-000000@bounces.bn.foxnews.com  Fri Apr 11 13:12:36 2025
 Subject: BREAKING NEWS: WATCH LIVE: White House holds briefing as tariff
  Folder: spamFolder                                                      10767
[ece404m6@shay ~/Mail]$ 
```

This logfile demonstrates the successful configuration of my spam filter account. As shown in the entries, the system correctly processes incoming emails according to the rules defined in the .procmailrc file. The first entry shows my test email being forwarded to my Purdue email address, while subsequent entries show various newsletter and news alert emails being sorted into the spamFolder. This confirms that both the Mail directory and the procmail configuration are working as expected, successfully filtering incoming messages based on their content and headers.