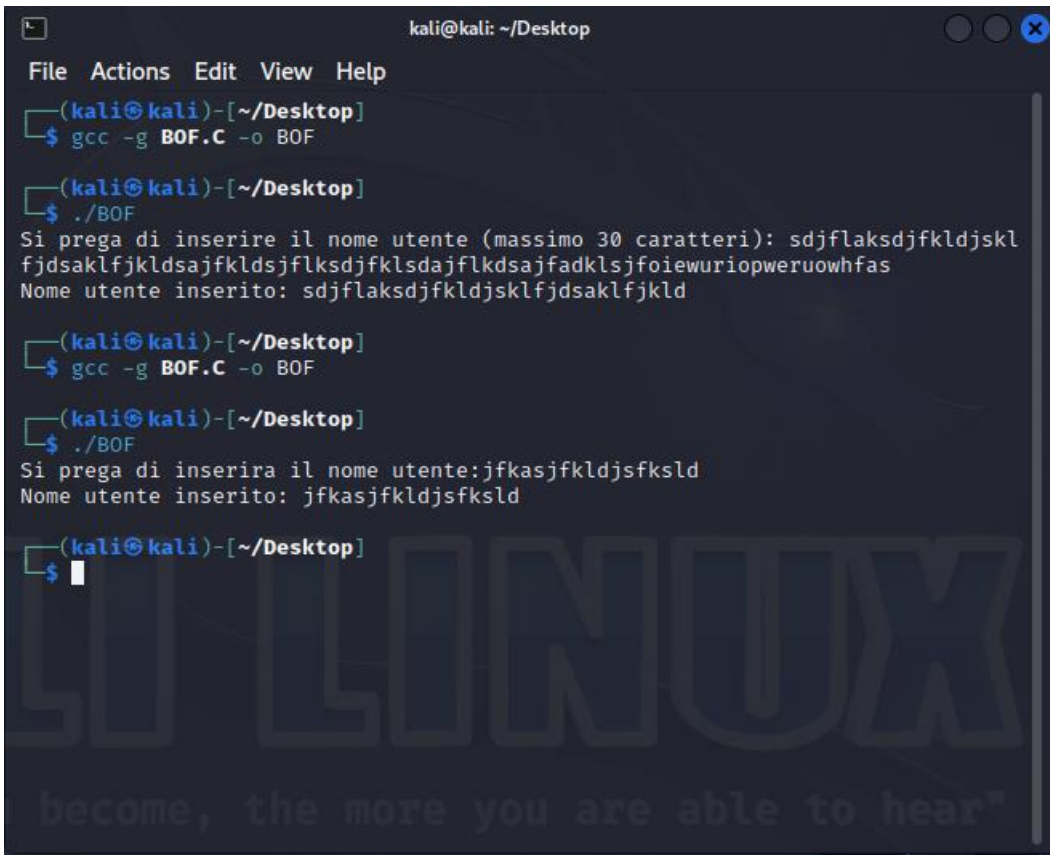


Buffer overflow

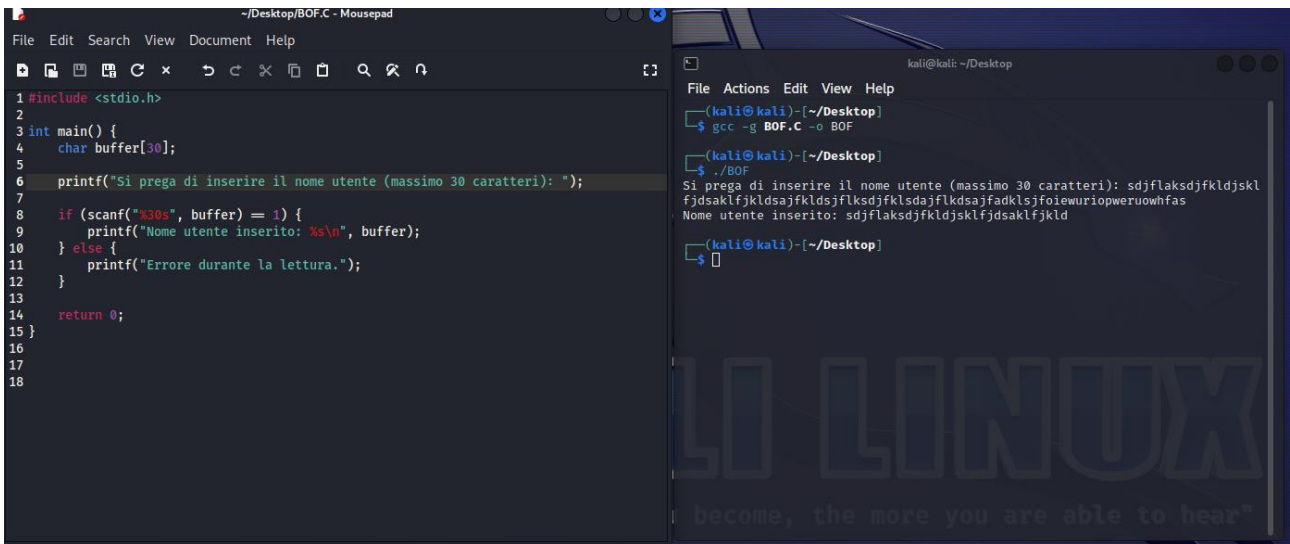
Oggi andremo a fare pratica con un programma in C. Il programma prevede il semplice inserimento di un nome utente con un massimo di 10 caratteri. Se proviamo a superare questo limite, il programma ci restituirà un errore in quanto i caratteri in eccesso creeranno un buffer overflow.



A terminal window on a Kali Linux system. The user compiles a program named BOF.C into an executable named BOF. They then run the program. The program prompts the user to enter a username (maximum 30 characters). The user enters a long string of random characters, which causes a buffer overflow. The program then prompts the user to enter a username again, and the user enters a shorter string. The terminal output shows the program's prompts and the user's input.

```
kali@kali: ~/Desktop
File Actions Edit View Help
(kali@kali)~[~/Desktop]
$ gcc -g BOF.C -o BOF
(kali@kali)~[~/Desktop]
$ ./BOF
Si prega di inserire il nome utente (massimo 30 caratteri): sdjflaksdjfkldjskl
fjdsaklfjkl dsajfklds jflksdjfklsdajfklsdajfklsjfoiewuriopweruowhfas
Nome utente inserito: sdjflaksdjfkldjsklfjdsaklfjkl
(kali@kali)~[~/Desktop]
$ gcc -g BOF.C -o BOF
(kali@kali)~[~/Desktop]
$ ./BOF
Si prega di inserira il nome utente:jfkasjfkldjsfksld
Nome utente inserito: jfkasjfkldjsfksld
(kali@kali)~[~/Desktop]
$
```

Per ovviare a questo problema andremo a modificare leggermente il programma, in modo che i caratteri che eccedono il limite massimo dichiarato non vengano conteggiati per la scelta dell'username. Il risultato sarà il seguente:



A screenshot showing a C program source code in a text editor and its execution in a terminal. The source code defines a buffer of size 30 and uses scanf to read a username. The terminal shows the program being compiled and run, with the user entering a long string that causes a buffer overflow. The program then prompts the user to enter a username again, and the user enters a shorter string.

```
~/Desktop/BOF.C - Mousepad
File Edit Search View Document Help
1 #include <stdio.h>
2
3 int main() {
4     char buffer[30];
5
6     printf("Si prega di inserire il nome utente (massimo 30 caratteri): ");
7
8     if (scanf("%30s", buffer) == 1) {
9         printf("Nome utente inserito: %s\n", buffer);
10    } else {
11        printf("Errore durante la lettura.");
12    }
13
14    return 0;
15 }
16
17
18
```

```
kali@kali: ~/Desktop
File Actions Edit View Help
(kali@kali)~[~/Desktop]
$ gcc -g BOF.C -o BOF
(kali@kali)~[~/Desktop]
$ ./BOF
Si prega di inserire il nome utente (massimo 30 caratteri): sdjflaksdjfkldjskl
fjdsaklfjkl dsajfklds jflksdjfklsdajfklsdajfklsjfoiewuriopweruowhfas
Nome utente inserito: sdjflaksdjfkldjsklfjdsaklfjkl
(kali@kali)~[~/Desktop]
$
```

Ricordiamo che un attacco “buffer overflow” è un tipo di attacco in cui un programma malevolo tenta di scrivere dati che eccedono lo spazio di memoria ad esso dedicato. Questo potrebbe permettere a chi effettua l’attacco potrebbe arrivare a sovrascrivere parti di memoria che non dovrebbe essere in grado di toccare. Un attacco del genere, se ben eseguito, potrebbe portare l’attaccante a prendere il controllo dell’intero sistema.