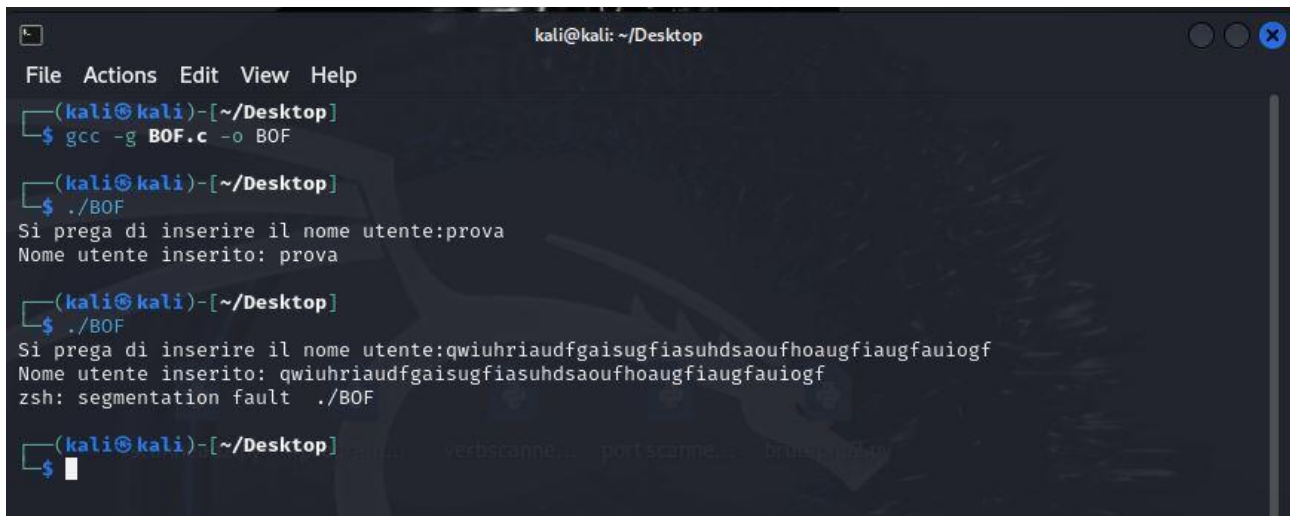


Testiamo il programma appena compilato:



```
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:prova
Nome utente inserito: prova
(kali@kali)~[~/Desktop]
$ ./BOF
Si prega di inserire il nome utente:qwihriaudfgaisugfiasuhdsoufhoaugfiaugfauiof
Nome utente inserito: qwihriaudfgaisugfiasuhdsoufhoaugfiaugfauiof
zsh: segmentation fault ./BOF
(kali@kali)~[~/Desktop]
$
```

Come vediamo inserendo un nome utente maggiore di 10 caratteri viene causato un Buffer overflow (zsh: segmentation fault).

Basta modificare il codice come segue per ovviare a questo problema:



```
BOF.c - Visual Studio Code
File Edit Selection View Go Run Terminal Help
C BOF.c
home > kali > Desktop > C BOF.c
1 #include <stdio.h>
2
3 int main () {
4
5 char buffer [30];
6
7 printf ("Si prega di inserire il nome utente:");
8 scanf ("%s", buffer);
9
10 printf ("Nome utente inserito: %s\n", buffer);
11
12 return 0;
13
14 }
```

Inserendo come capienza massima di caratteri il numero 30, ora il programma accetta nomi utenti fino appunto a 30 caratteri:

```
(kali@kali)-[~/Desktop]
$ gcc -g BOF.c -o BOF

(kali@kali)-[~/Desktop]
$ ./BOF
Si prega di inserire il nome utente:adshguytoppoklhtrfioosirwrnbmo
Nome utente inserito: adshguytoppoklhtrfioosirwrnbmo

(kali@kali)-[~/Desktop]
$
```

Ovviamente inserendo un nome utente con una lunghezza superiore a 30 viene nuovamente causato un buffer overflow:

```
(kali@kali)-[~/Desktop]
$ ./BOF
Si prega di inserire il nome utente:fsojff0ajfsjfsjfpsofdfposjfpasjdpasfjpsdjfpaofapjsa
Nome utente inserito: fsojff0ajfsjfsjfpsofdfposjfpasjdpasfjpsdjfpaofapjsa
zsh: segmentation fault ./BOF
```