Scriviamo un semplice codice che soffre della vulnerabilità di **buffer overflow**.

```c
#include <stdio.h>

int main () {

char buffer [10];

printf ("Si prega di inserire il nome utente:");
scanf ("%s", buffer);

printf ("Nome utente inserito: %s\n", buffer);

return 0;

}
```

Compiliamo il file utilizzando il comando **gcc –g BOF.c –o BOF**; una volta fatto eseguiamo il codice con **./BOF**.

```
Si prega di inserire il nome utente:qwertyuiopasdfghjklzxcvbnmqwer
Nome utente inserito: qwertyuiopasdfghjklzxcvbnmqwer
zsh: segmentation fault  ./BOF

┌──(kali㉿kali)-[~/Desktop]
└─$
```

Se inseriamo più di 10 caratteri il programma ci ritorna l'errore **segmentation fault**, ovvero errore di segmentazione; esso avviene quando un programma tenta di scrivere contenuti su una porzione di memoria alla quale non ha accesso. Infatti abbiamo inserito 30 caratteri in un buffer che ne può contenere 10 e di conseguenza alcuni caratteri stanno sovrascrivendo aree di memorie inaccessibili, causando un **crash**.