Abbiamo già parlato del buffer overflow, una vulnerabilità che è conseguenza di una mancanza di controllo dei limiti dei buffer che accettano input utente. Nelle prossime slide vedremo un esempio di codice in C volutamente vulnerabile ai BOF, e come scatenare una situazione di errore particolare chiamata «segmentation fault», ovvero un errore di memoria che si presenta quando un programma cerca inavvertitamente di scrivere su una posizione di memoria dove non gli è permesso scrivere (come può essere ad esempio una posizione di memoria dedicata a funzioni del sistema operativo).
Riportiamo il seguente codice su kali linux

```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;

}

```

Dopo averlo riportato andate su kali e digitate ./BOF per farlo partire e mettete un nome utente che sia minore di 10 caratteri.

```
┌──(kali㉿kali)-[~/Desktop]
└─$ ./BOF
Si prega di inserire il nome utente:kali
Nome utente inserito: kali
```

Se invece mettete un nome utente maggiore di 10 caratteri vi darà questa risposta.

```
└─$ ./BOF
Si prega di inserire il nome utente:qqrsfsyshsudtsbxysjsgs
Nome utente inserito: qqrsfsyshsudtsbxysjsgs
zsh: segmentation fault  ./BOF
```

E quindi per mettere un nome utente più di 10 caratteri bisogna modificare il codice, che in questo caso non funziona comunque con un nome utente più di 10 caratteri.