Buffer Overflow

In Kali creiamo il seguente script in C e lo salviamo come BOF.c nella directory `/Esercizi/BufferOverflow`. Il rpogramma permette di inserire il nome in una variabile "buffer" di lunghezza 10 e poi lo stampa a video:

```
 GNU nano 7.2
#include <stdio.h>

int main() {

char buffer[10];

printf ("Si prega di inserire il nome utente: ");

scanf("%s", buffer);

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

return 0;

}
```

Dalla directory che contiene il file compiliamo lo script con il comando `gcc BOF.c -o BOF` :

```
┌──(kali㉿kali)-[~/Esercizi/BufferOverflow]
└─$ gcc BOF.c -o BOF
```

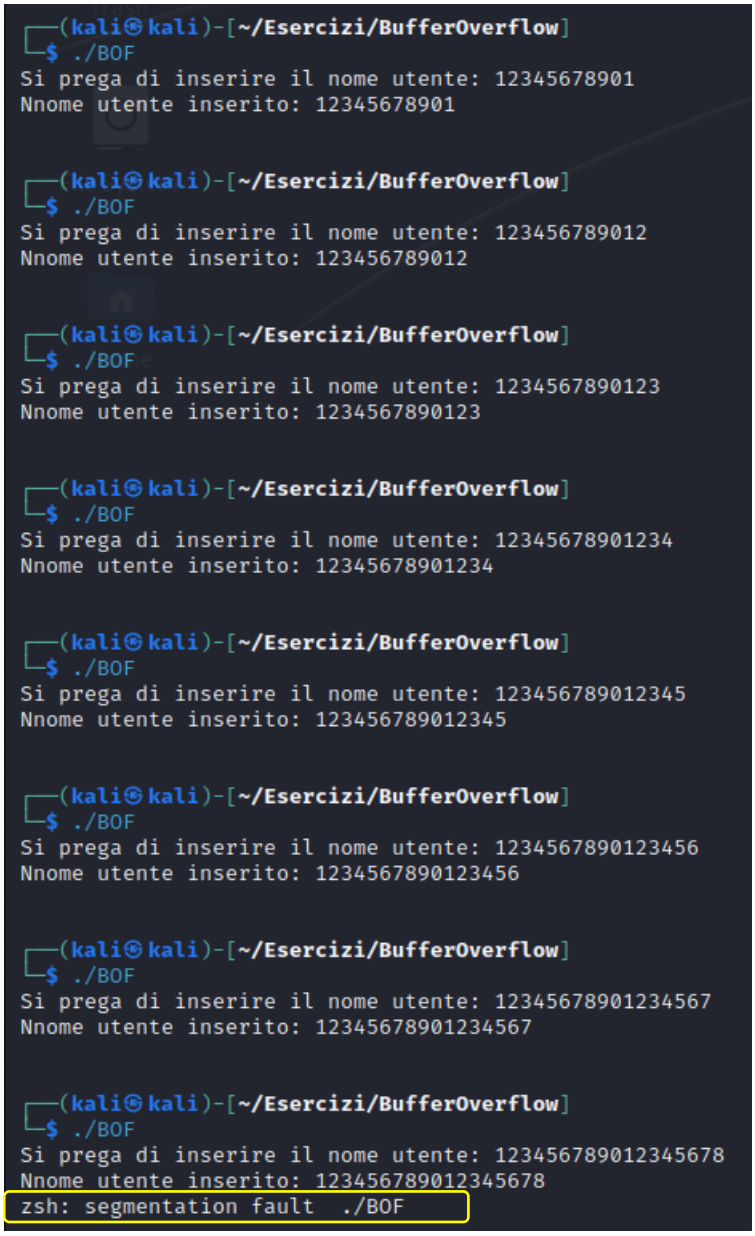
Eseguiamo poi lo script con il comando `./BOF` (il nome `BOF` per l'esecuzione è quello specificato in fase di compilazione dopo lo switch `-o`) :

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

Il programma si avvia chiedendoci un nome utente. Inserendo un nome di 6 caratteri il programma si comporta normalmente:

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

Nel codice C la variabile del nome ha lunghezza 10. Provando ad inserire stringhe progressivamente più lunghe, vediamo che il programma segnala un errore di buffer overflow quando inseriamo una stringa di lughezza 18:



L'errore di segmantazione avviene perché il programma tenta di scrivere contenuti in prozioni di memoria che non gli sono state assegnate.

Modifichiamo il programma incrementando la lunghezza della variabile a 30:

```c
#include <stdio.h>

int main() {

char buffer[30];

printf ("Si prega di inserire il nome utente: ");

scanf("%s", buffer);

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

return 0;

}
```

Compiliamo nuovamente il programma ed esguiamolo. Inserendo una stringa di 50 caratteri l'errore di segmentazione si ripresenta:

```
┌──(kali㉿kali)-[~/Esercizi/BufferOverflow]
└─$ ./BOF
Si prega di inserire il nome utente: sdjfjsbjbvkjnvkjsbkjbkjfdbkdbkvbdknvfbvdfjbvjkbvjb
Nome utente inserito: sdjfjsbjbvkjnvkjsbkjbkjfdbkdbkvbdknvfbvdfjbvjkbvjb
zsh: segmentation fault  ./BOF
```

Riproviamo con stringhe di lungehzza, rispettivamente 37, 40 e 39. Dalla risposta del programma vediamo che il limite di lunghezza minimo per generare l'errore in questo caso è 40:

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


┌──(kali㉿kali)-[~/Esercizi/BufferOverflow]
└─$ ./BOF
Si prega di inserire il nome utente: sdjfjsbjbvkjnvkjsbkjbkjfdbkdbkvbdknvfefg
Nome utente inserito: sdjfjsbjbvkjnvkjsbkjbkjfdbkdbkvbdknvfefg
zsh: segmentation fault  ./BOF

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