CONSEGNA U2 S7 L4

Per svolgere l'esercizio, la prima cosa che faccio è copiare il codice necessario a testare il **Buffer Overflow**. Questo tipo di vulnerabilità può essere sfruttata da un malintenzionato per eseguire del codice malevolo che viene salvato in memoria in maniera abusiva. Si può manifestare quando ad esempio, viene inserito un input da parte dell'utente che supera la grandezza massima che può essere contenuta all'interno di una variabile, ed il contenuto della suddetta va a riempire un numero indefinito di celle di memoria oltre quelle ad essa riservate.

In questo esercizio proverò ad ottenere questo risultato causando forzatamente l'errore di **segmentation fault** con l'aiuto di un piccolo programmino scritto in C.

```
File  Actions  Edit  View  Help
 GNU nano 8.0                                  BOF.c
#include <stdio.h>

int main(){
        char buffer [10];
        printf("Inserire il nome utente:");
        scanf("%s", buffer);
        printf("Il nome che hai inserito è: %s\n", buffer);
}
```

In secondo luogo, eseguo la compilazione dello stesso con il compilatore gcc. Il comando che ho usato: **gcc nome_file.c -o file_eseguibile.**

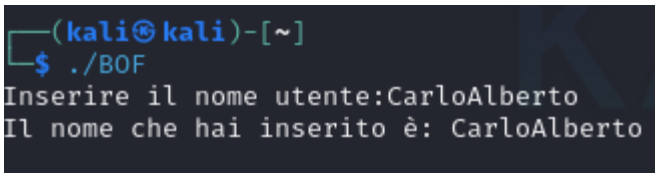
```
┌──(kali㉿kali)-[~]
└─$ gcc BOF.c -o BOF
```

Inizio verificando il corretto funzionamento dell'eseguibile. Provo a digitare un nome.

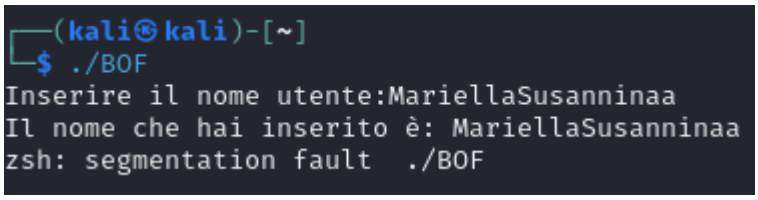
```
┌──(kali㉿kali)-[~]
└─$ ./BOF
Inserire il nome utente:Gianluca
Il nome che hai inserito è: Gianluca
```

Il nome contiene *8 caratteri*, quindi è perfettamente contenuto all'interno del vettore di *10 caratteri*.

Poi provo con un nome lungo *12 caratteri,* che eccede di ben due caratteri la lunghezza massima del vettore all'interno del codice. Il programma non va in crash e non presenta alcun errore, nonostante sia stato superato il limite massimo di caratteri memorizzabili.



Dopo diversi tentativi, raggiunto il limite dei *18 caratteri* il programma presenta l'errore di **segmentation fault**.



Cambiando il numero massimo di celle del vettore a *30 caratteri*, provo a fare diversi tentativi anche questa volta con nomi di lunghezza differente, fino a capire il limite massimo di caratteri entro il quale non si presenta alcun errore.

L’errore si verificherà solo dopo aver superato i *39 caratteri*.