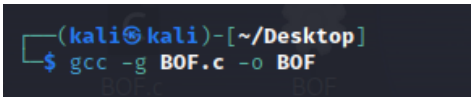
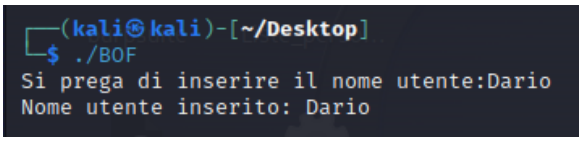
Esercizio 09-03

Traccia: testare un codice C volutamente vulnerabile ai BOF e 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).

Creazione del codice vulnerabile a BOF



Compilazione del codice



Avvio del programma

Un input <10 caratteri non crea nessun problema

Con un input >10 il programma restituisce un errore <<segmentation fault>>

kali@kali: ~/Desktop

File Actions Edit View Help

GNU nano 7.2 BOF.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;


}
```

Modifica del codice aumentando il buffer da 10 a 30 caratteri

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

Compilazione del codice modificato

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

Un input >10 caratteri non restituisce più problemi

Un input >30 caratteri restituisce sempre lo stesso errore <<segmentation fault>>

```
┌──(kali㉿kali)-[~/Desktop]
└─$ ./BOF
Si prega di inserire il nome utente:QualoraTeLoFossiDimenticatoIoSoCheSopportiF
inoATrentaCaratteriEQuindiScriveròFincheNonMiRestituiraiQuellErroreCheStoCercan
doDiOttenere
Nome utente inserito: QualoraTeLoFossiDimenticatoIoSoCheSopportiFinoATrentaCara
tteriEQuindiScriveròFincheNonMiRestituiraiQuellErroreCheStoCercandoDiOttenere
zsh: segmentation fault  ./BOF
```