

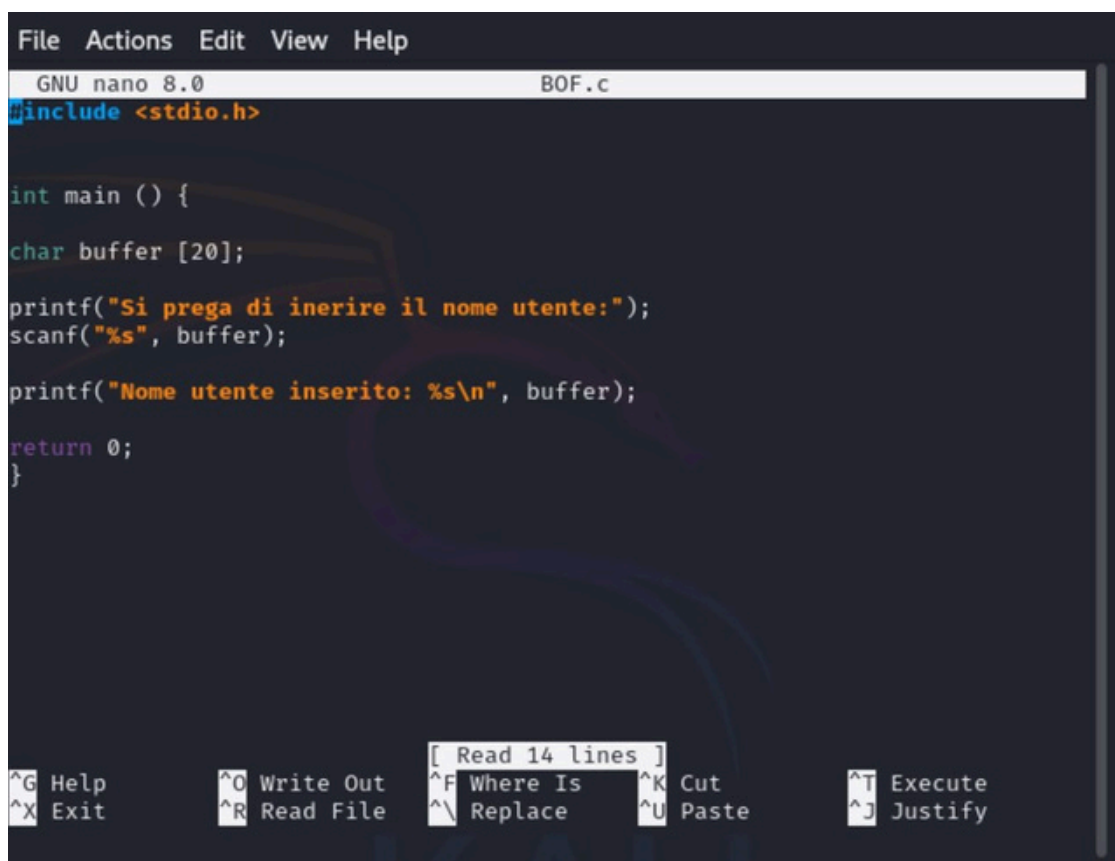
BUFFER OVERFLOW

Traccia

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).

Mi sposto sul desktop e procedo tramite il comando **sudo nano BOF.c** a scrivere il programma come richiesto dalla traccia

```
(kali㉿kali)-[~]  
$ cd Desktop  
  
(kali㉿kali)-[~/Desktop]  
$ sudo nano BOF.c
```



```
File Actions Edit View Help  
GNU nano 8.0 BOF.c  
#include <stdio.h>  
  
int main () {  
    char buffer [20];  
    printf("Si prega di inserire il nome utente:");  
    scanf("%s", buffer);  
    printf("Nome utente inserito: %s\n", buffer);  
    return 0;  
}
```

[Read 14 lines]
^G Help ^O Write Out ^F Where Is ^K Cut ^T Execute
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify

A questo punto procedo a compilare il file come richiesto utilizzando il comando

gcc -g BOF.c -o BOF

N.B ad ogni modifica del codice il programma andrà ricompilato

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

Inserendo un nome utente di 8 caratteri(Maurizio) , il programma non ci riporta nessun problema, infatti come sappiamo il buffer accetta fino a 10 caratteri

```
(kali㉿kali)-[~]
$ cd Desktop

(kali㉿kali)-[~/Desktop]
$ sudo nano BOF.c
[sudo] password for kali:

(kali㉿kali)-[~/Desktop]
$ y
y: command not found

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

(kali㉿kali)-[~/Desktop]
$
```

100

```
(kali㉿kali)-[~/Desktop]
$ ./BOF
Si prega di inserire il nome utente:mamamamamamamamamamamamamaamamamamam
amamamamama
Nome utente inserito: mamamamamamamamamamamamamaamamamamamamamamamama
zsh: segmentation fault ./BOF
```

Procedo così a modificare il codice del programma aumentando il limite di caratteri accettati a 30.

The screenshot shows a terminal window with the title 'kali@kali: ~/Desktop'. The nano editor is open, editing a file named 'BOF.c'. The code in the file is as follows:

```

#include <stdio.h>

int main () {
    //aumento la dimensione del vettore a 30
    char buffer [20];

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

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

    return 0;
}

```

The user has just typed 'include <stdio.h>' on the first line. The terminal shows the standard nano editor interface with a menu bar at the top and a status bar at the bottom. The status bar indicates 'Read 14 lines'.

Inserisco un input di 30 caratteri e noto che non ricevo piu l'errore

[illegible]

100

```
(kali㉿ kali)-[~/Desktop]
$ ./BOF
Si prega di inserire il nome utente:iiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiii
iiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiii
Nome utente inserito: iiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiii
iiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiii
zsh: segmentation fault  ./BOF

(kali㉿ kali)-[~/Desktop]
$
```