

Buffer overflow

Il buffer overflow è una vulnerabilità presente in un codice, derivata dalla mancanza di controllo dei limiti dei buffer (una regione di memoria temporanea utilizzata per immagazzinare dati) che accettano input dell'utente. Ne scaturisce che tale vulnerabilità può mandare in tilt la CPU a tal punto da farle eseguire del codice malevolo.

Questo è un esempio di codice in linguaggio C volutamente vulnerabile ai BOF, che accetta in input il nome di un utente ed ha una memoria buffer di 10:

```
GNU nano 6.3
#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;

}
```

Andremo a 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).

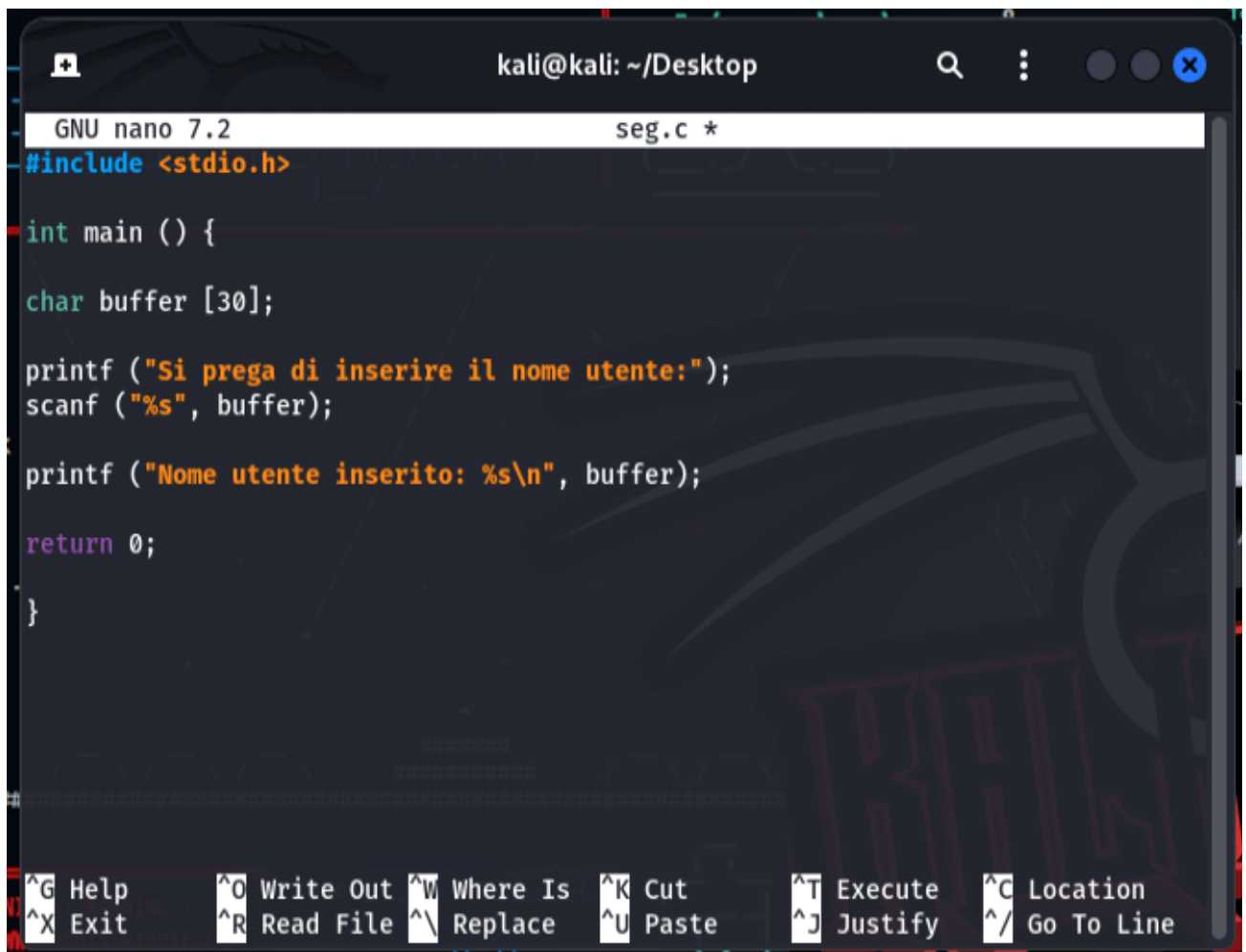
Se inseriamo un nome di 5 caratteri, come ad esempio "Mario" notiamo che non succede nulla:

```
kali@kali: ~/Desktop
(kali@kali)-[~/Desktop]
$ nano seg.c
(kali@kali)-[~/Desktop]
$ gcc -g seg.c -o seg
(kali@kali)-[~/Desktop]
$ ./seg
Si prega di inserire il nome utente:Mario
Nome utente inserito: Mario
```

Mentre se inseriamo una sequenza più lunga ci ridà l'errore:

```
(kali@kali)-[~/Desktop]
$ ./seg
Si prega di inserire il nome utente:qwertyuiopasdfghjklzxcvbnm
Nome utente inserito: qwertyuiopasdfghjklzxcvbnm
zsh: segmentation fault (core dumped) ./seg
```

Se modifichiamo quindi il buffer nel codice, portandolo a 30:



```
kali@kali: ~/Desktop
GNU nano 7.2 seg.c *
#include <stdio.h>

int main () {

char buffer [30];

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

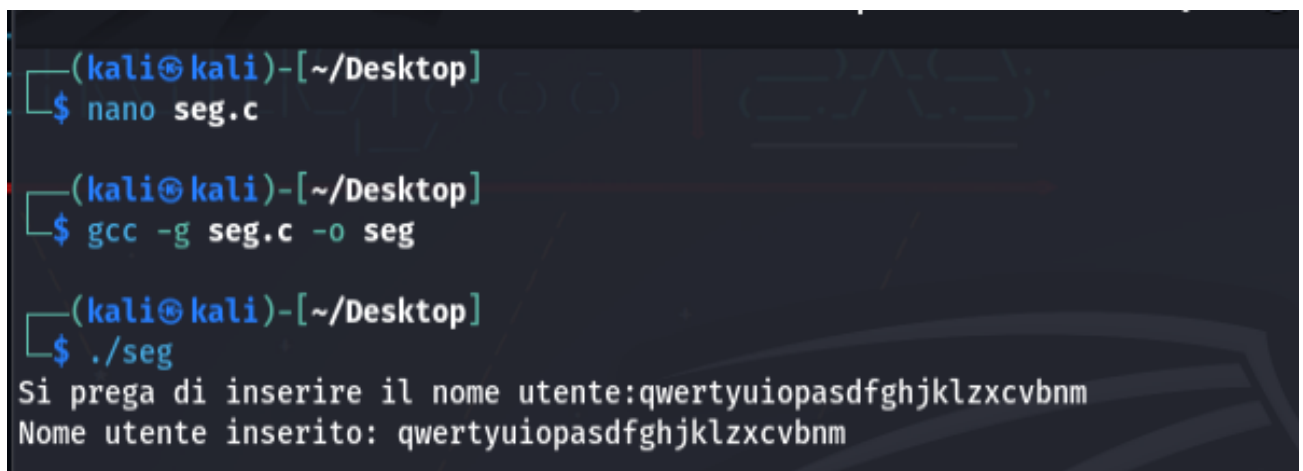
printf ("Nome utente inserito: %s\\n", buffer);

return 0;

}

^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File ^_ Replace   ^U Paste     ^J Justify   ^_ Go To Line
```

e riproviamo:



```
(kali@kali)-[~/Desktop]
$ nano seg.c

(kali@kali)-[~/Desktop]
$ gcc -g seg.c -o seg

(kali@kali)-[~/Desktop]
$ ./seg
Si prega di inserire il nome utente:qwertyuiopasdfghjklzxcvbnm
Nome utente inserito: qwertyuiopasdfghjklzxcvbnm
```

possiamo vedere che non da più l'errore in quanto la memoria supporta più caratteri.