

## Report sul Buffer Overflow

### Introduzione

L'esercizio ha avuto come obiettivo la comprensione del concetto di buffer overflow, ho scritto e testato un codice in linguaggio C che richiede all'utente di inserire un nome, durante l'esperimento, ho variato la dimensione del buffer per osservare gli effetti dell'overflow.

### Codice Sviluppato

Il codice sviluppato richiede all'utente di inserire un nome e lo stampa, ho inizialmente impostato un buffer di 10 caratteri, poi ho modificato la dimensione del buffer a 30 caratteri.

```
kali@kali: ~/Desktop
File Actions Edit View Help
GNU nano 7.2 buffer.c
#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;
}
```

```
kali@kali: ~/Desktop
File Actions Edit View Help
GNU nano 7.2 buffer.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;
}
```

## Primo Test: Buffer di 10 Caratteri

Nel primo test, il buffer era impostato a 10 caratteri.

```
kali@kali: ~/Desktop
File Actions Edit View Help
zsh: corrupt history file /home/kali/.zsh_history
(kali@kali)-[~]
$ pwd
/home/kali

(kali@kali)-[~]
$ cd /home/kali/Desktop

(kali@kali)-[~/Desktop]
$ ./buffer
si prega di inserire il nome utente:luca
nome utente inserito: luca

(kali@kali)-[~/Desktop]
$
```

Siccome il buffer impostato a 10 permette di inserire un massimo di 10 caratteri in questo caso non ha dato errori.

```
kali@kali: ~/Desktop
File Actions Edit View Help
zsh: corrupt history file /home/kali/.zsh_history
(kali@kali)-[~]
$ pwd
/home/kali

(kali@kali)-[~]
$ cd /home/kali/Desktop

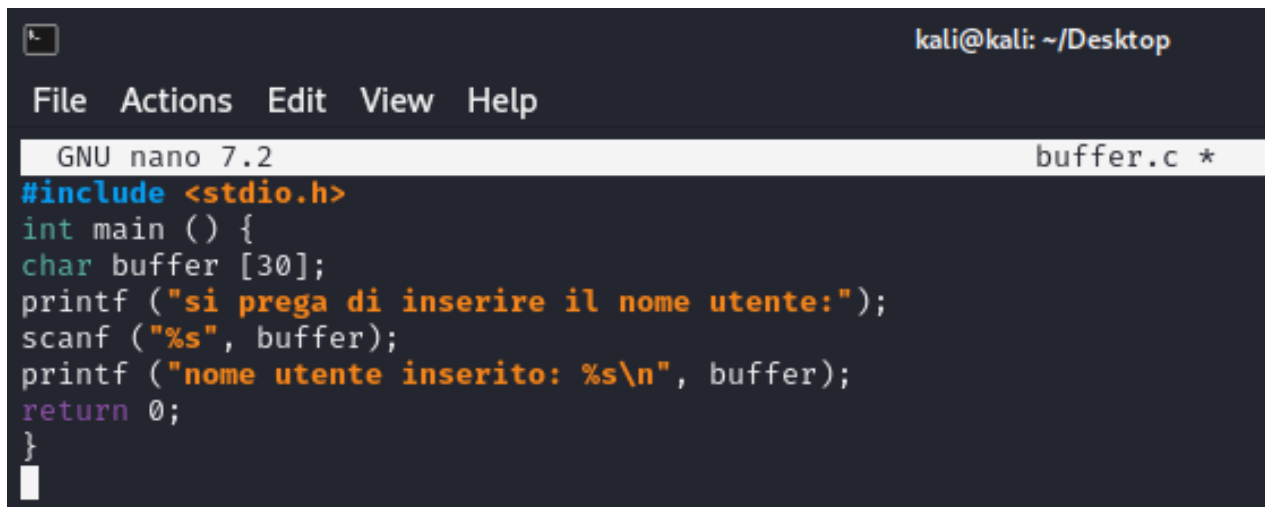
(kali@kali)-[~/Desktop]
$ ./buffer
si prega di inserire il nome utente:luca
nome utente inserito: luca

(kali@kali)-[~/Desktop]
$ ./buffer
si prega di inserire il nome utente:ciaomichiamolucaho24annievangodaascolipiceno
nome utente inserito: ciaomichiamolucaho24annievangodaascolipiceno
zsh: segmentation fault ./buffer
```

In questo altro caso inserendo più di 10 caratteri il programma ha restituito un errore, questi caratteri in più hanno causato un BOF.

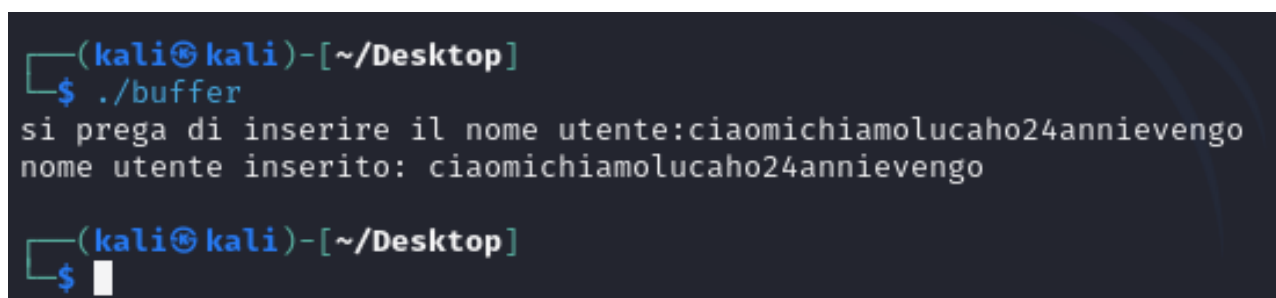
## Secondo Test: Buffer di 30 Caratteri

Nel secondo test, ho modificato il buffer a 30 caratteri.



```
kali@kali: ~/Desktop
File Actions Edit View Help
GNU nano 7.2 buffer.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;
}
```

Con questa modifica, il programma ha gestito correttamente l'input più lungo senza causare un buffer overflow, anche inserendo una stringa lunga 29 caratteri, il programma ha funzionato correttamente, stampando l'intero input senza errori.



```
(kali@kali)-[~/Desktop]
$ ./buffer
si prega di inserire il nome utente:ciaomichiamolucaho24annievengo
nome utente inserito: ciaomichiamolucaho24annievengo

(kali@kali)-[~/Desktop]
$
```

Per la compilazione e l'esecuzione del programma ho usato i comandi: **gcc buffer.c -o buffer**

**./buffer**

Questo esercizio ha dimostrato come un buffer di dimensioni insufficienti possa portare a un buffer overflow, causando errori di segmentazione e potenziali vulnerabilità di sicurezza.

Incrementando la dimensione del buffer, il programma ha gestito correttamente input più lunghi senza errori.