S7/L4

BUFFER OVERFLOW

Colombo Federico

Nell'esercizio di oggi andremo a spiegare cos'è lo stack overflow e come risolvere un semplice programma in C.

Lo stack overflow è un errore comune che si verifica quando un programma utilizza più spazio di quello disponibile nello stack della memoria.
Lo stack è una regione di memoria che viene utilizzata per memorizzare le variabili locali e i dati relativi alle chiamate di funzione.
Quando una funzione viene chiamata, i suoi parametri e altre informazioni vengono inseriti nello stack. Tuttavia, se il programma utilizza in modo eccessivo lo stack, si verifica lo stack overflow.

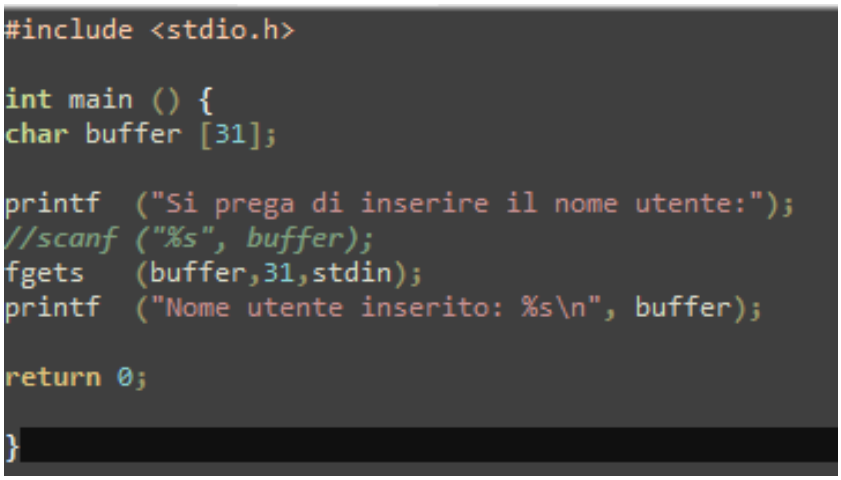
Vediamo l'esempio del programma di oggi:

```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;

}
```

Questo programma chiede all'utente di inserire un nome utente e quindi stampa il nome utente inserito. Tuttavia, il problema è che il buffer ha dimensione fissa di 10 caratteri. Ciò significa che se l'utente inserisce una stringa più lunga di 9 caratteri, si verificherà un buffer overflow, che può causare comportamenti imprevisti o addirittura vulnerabilità di sicurezza nel programma.

Risoluzione

Per risolvere il problema del buffer overflow, si può utilizzare la funzione **fgets** per leggere l'input dell'utente invece della funzione **scanf**.
Inoltre possiamo aumentare la quantità di caratteri che l'utente può inserire nel seguente modo:



```c
#include <stdio.h>

int main () {
char buffer [31];

printf  ("Si prega di inserire il nome utente:");
//scanf ("%s", buffer);
fgets   (buffer,31,stdin);
printf  ("Nome utente inserito: %s\n", buffer);

return 0;

}
```

Così facendo l'utente può immettere un massimo di 30 caratteri senza che il nostro programma vada in crash.