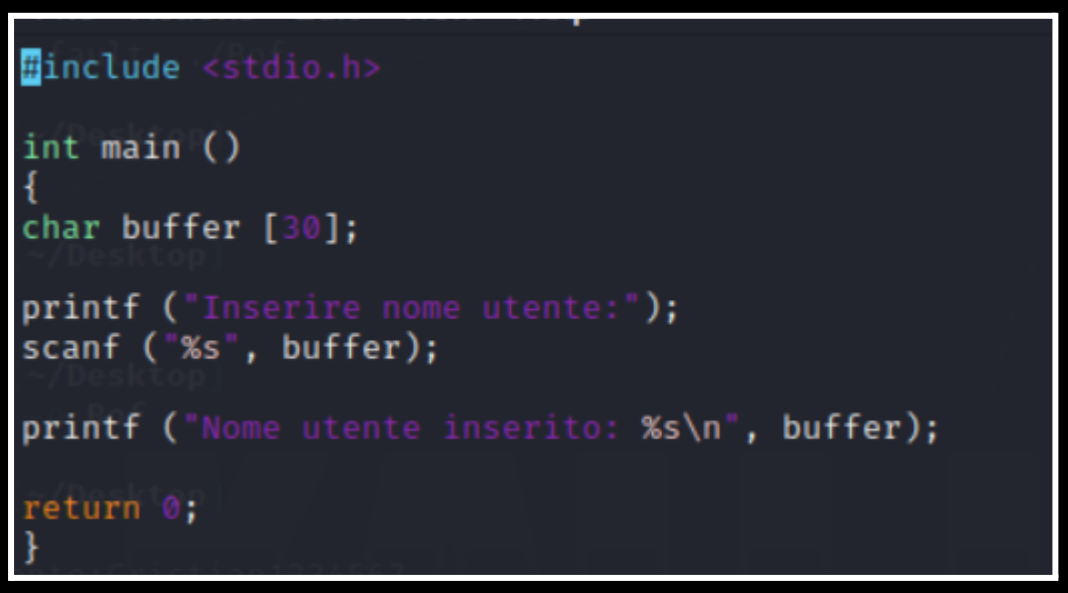
Buffer Overflow

Richiesta: Risolvi la vulnerabilità BOF del seguente programma:

Aumentando la dimensione del vettore a 30.

Esecuzione

Per evitare questa vulnerabilità di tipo BOF (Buffer Overflow), aumentiamo la dimensione del vettore a 30, rendendo più sicuro l'input dell'utente:



Adesso testiamo il programma e osserviamo come si comporta inserendo più di 10 caratteri nell'input:

Il programma non ritorna nessun errore di "segmentation fault", ovvero un errore si segmentazione.
Questo tipo di errore avviene quando un programma tenta di scrivere contenuti su una porzione di memoria alla quale non ha accesso, di conseguenza si sviluppa una vulnerabilità di BOF.

Sicurezza
Con il metodo eseguito, aumentando il vettore a 30 come richiesto, non risolviamo del tutto la vulnerabilità.
Se un utente digitasse più di 30 caratteri, la memoria buffer riservata all'input andrebbe a straripare, sovrascrivendo di conseguenza altre parti di memoria.

Per risolvere definitivamente questa vulnerabilità, impostiamo il limite massimo di caratteri che il programma andrà a leggere:

```c
#include <stdio.h>

int main ()
{
char buffer [10];

printf ("Inserire nome utente:");
scanf ("%9s", buffer);

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

return 0;
}
```

Usando "%9s", ci assicuriamo che scanf non scriva più di 29 caratteri nel buffer, lasciando spazio per il carattere nullo \0 che indica la fine della stringa in C.
Questo previene il segmentation fault perché evita di sovrascrivere la memoria oltre il buffer.