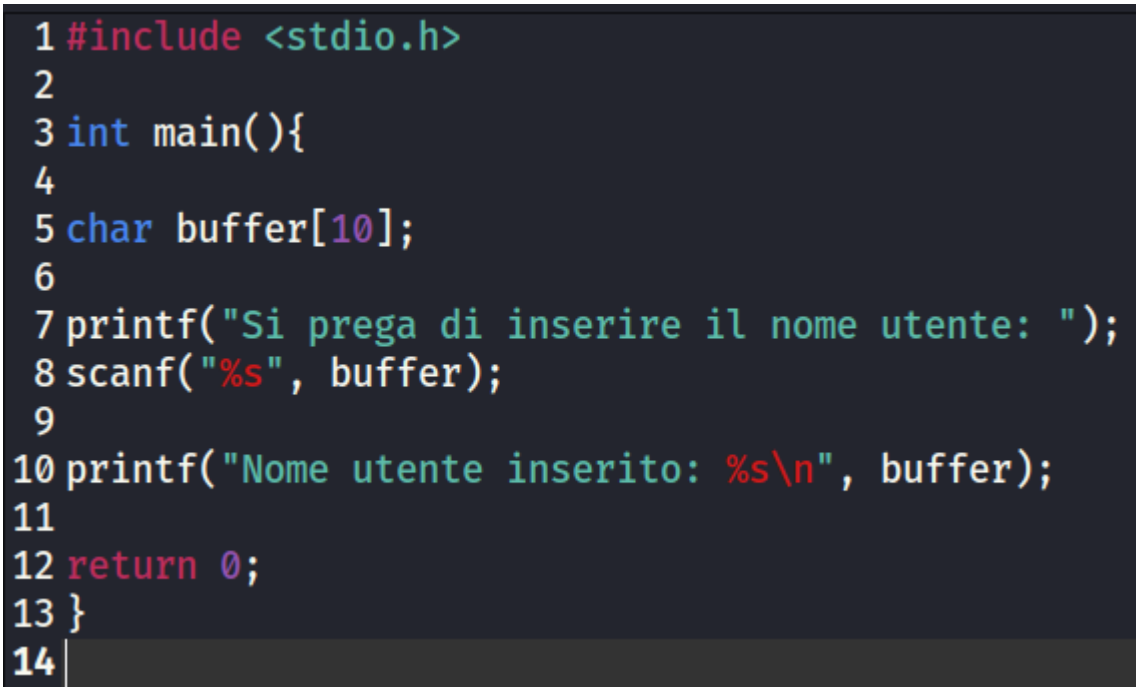
Epicode
CS-0124
Francesco Ficetti
Pratica S7/L4

Indice

Traccia

Dato il seguente programma, modificare la dimensione del vettore a 30.



```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;
}

```

Svolgimento

Modifica del programma

Il buffer overflow è una condizione di errore che si verifica quando un programma cerca di scrivere in un’area di memoria alla quale non ha accesso. Nel nostro caso, il programma è stato modificato come da figura.

```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;
}

```

Esecuzione del programma

Essendo il C un linguaggio compilato, il programma deve essere passato come parametro ad un compilatore, che lo controlla e ne crea un file eseguibile. Il comando da eseguire è *gcc nomefile.c*, dove *nomefile* va sostituito con il nome del file che si vuole compilare.
Per eseguire il file, il comando è *./a.out*.

```
┌──(francesco㉿kali)-[~/Desktop/Workspace]
└─$ ./a.out
Si prega di inserire il nome utente: aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
Nome utente inserito: aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
zsh: segmentation fault  ./a.out
```

Come si vede dalla figura, in input è stata data una stringa più lunga di 30 caratteri, che sono quelli accettati dall’array. Il programma cercherà quindi di scrivere in un’area di memoria esterna da quella ad esso dedicata, questa azione restituirà l’errore di tipo *segmentation fault*.

Conclusione

L’attacco è avvenuto con successo, perché il programma non effettua nessun controllo sull’input dell’utente.