

# Esercitazione WEEK 17 D4

## Buffer Overflow

Ettore Farris

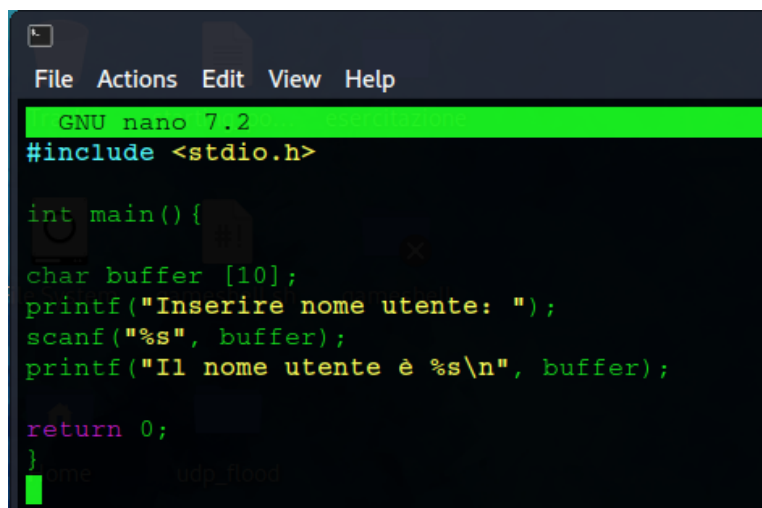
### Descrizione sintetica

In questa esercitazione vedremo un semplice esempio di codice in C volutamente vulnerabile ai BOF, e come 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).

### Svolgimento

- Codice C

Il codice C è un semplice programma che prende l'input da un utente e lo stampa a video. I caratteri inseriti vengono immagazzinati in un array di un 10 di caratteri. Eccedendo questo numero, inserendone ad esempio 15 anziché 10, verrà restituito l'errore *segmentation fault*, che sta alla base degli attacchi buffer overflow.




```
GNU nano 7.2
#include <stdio.h>

int main() {
    char buffer [10];
    printf("Inserire nome utente: ");
    scanf("%s", buffer);
    printf("Il nome utente è %s\n", buffer);

    return 0;
}
```

Compiliamo il codice col comando `gcc nome.c -o nome` e lanciamo l'eseguibile col comando `./nome`.



```
(kali㉿kali) - [~]  
$ ./nome  
Inserire nome utente: Ettore  
Il nome utente è Ettore  
  
(kali㉿kali) - [~]  
$ ./nome  
Inserire nome utente: hfoihoifhsdoifjsdoifhsofhsfijdsiofhdsiofhdsiofihsfoihdsfoihdsfiidshfosd  
zsh: segmentation fault ./nome  
  
(kali㉿kali) - [~]  
$
```

I caratteri in più che sono stati inseriti “invadono” quindi gli spazi di memoria contigui sovrascrivendoli.