Consegna S7-L4

Buffer Overflow

Traccia

Provate a riprodurre l’errore di segmentazione modificando il programma come di seguito:

Aumentando la dimensione del vettore a 30;

Fare la prova dell’errore

modificare il codice in modo che l’errore non si verifichi (es aumentare il vettore a 30 o fare dei controlli)

Verificare, modificando il codice, dove va a scrivere i caratteri in overflow

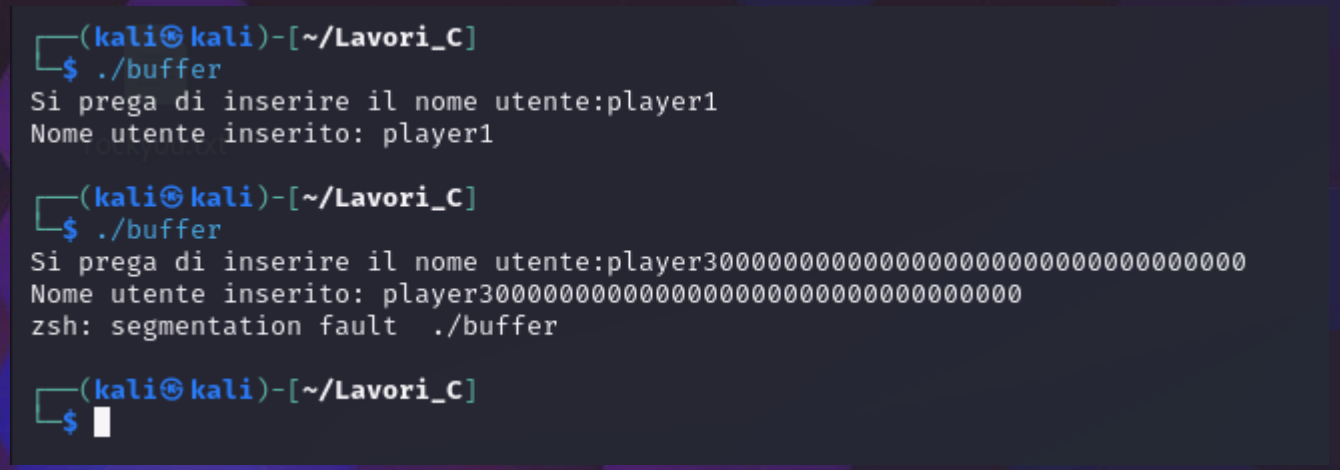
Codice iniziale

Il codice iniziale dato dalla traccia presenta la dimensione del buffer a 10.

```
  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;

}
```

Errore nel buffer

Se viene inserito un nome utente inferiore ai caratteri consentiti, tutto è corretto. Tuttavia se l’utente inserisce un nome superiore alla dimensione dell’array, avviene un avviso di segmentation fault.



Modifica della dimensione

Ho aumentato il buffer a 30 caratteri e implementato una variabile e il comando getchar per pulire il buffer all'inserimento del carattere \n una volta inserito il nome utente desiderato o nel caso si arriva al limite massimo di caratteri. In questo modo si evita di lasciare in buffer caratteri in eccesso. Nella seconda immagine si può notare come i caratteri in eccesso vengono tagliati e prende in input solo i primi 29.

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

int main () {

char buffer[30];

printf ("Si prega di inserire il nome utente:");
scanf ("%29s", buffer);

// inseirimento di un ciclo per pulire il buffer

int c;
while (( c = getchar()) != '\n' && c != EOF);

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

return 0;

}
```

```
┌──(kali㉿kali)-[~/Lavori_C]
└─$ gcc -g buffer.c -o buffer

┌──(kali㉿kali)-[~/Lavori_C]
└─$ ./buffer
Si prega di inserire il nome utente:amedeoooooooooooooooooooooooooooooooooooooooooooooooo
Nome utente inserito: amedeooooooooooooooooooooooooo

┌──(kali㉿kali)-[~/Lavori_C]
└─$
```

Cosa succede con un buffer overflow

In questo esempio, ho inserito una variabile int che va a sovrascriversi nel caso in cui l'utente vada ad inserire nel buffer più caratteri di quelli consentiti.

```
┌──(kali㉿kali)-[~/Lavori_C]
└─$ ./buffer2
Prima dell'overflow: variabileImportante = 42
Si prega di inserire il nome utente: amedeo
Nome utente inserito: amedeo
Dopo l'overflow: variabileImportante = 42

┌──(kali㉿kali)-[~/Lavori_C]
└─$ ./buffer2
Prima dell'overflow: variabileImportante = 42
Si prega di inserire il nome utente: amedeogagawrg
Nome utente inserito: amedeogagawrg
Dopo l'overflow: variabileImportante = 6779511

┌──(kali㉿kali)-[~/Lavori_C]
└─$ ./buffer2
Prima dell'overflow: variabileImportante = 42
Si prega di inserire il nome utente: amedeo56
Nome utente inserito: amedeo56
Dopo l'overflow: variabileImportante = 42

┌──(kali㉿kali)-[~/Lavori_C]
└─$ ./buffer2
Prima dell'overflow: variabileImportante = 42
Si prega di inserire il nome utente: amedeo64462223
Nome utente inserito: amedeo64462223
Dopo l'overflow: variabileImportante = 858927666

┌──(kali㉿kali)-[~/Lavori_C]
└─$
```

```
  GNU nano 7.2                                  buffer2.c
#include <stdio.h>

int main() {
    char buffer[10];  // Array con capacità per 5 caratteri
    int variabileImportante = 42;

    printf("Prima dell'overflow: variabileImportante = %d\n", variabileImportante);

    printf("Si prega di inserire il nome utente: ");
    scanf("%s", buffer);  // Attenzione: nessun controllo sulla dimensione dell'array

    // Stampa il contenuto dell'array buffer
    printf("Nome utente inserito: %s\n", buffer);

    // Stampa l'indirizzo di memoria e il valore della variabile dopo l'overflow
    printf("Dopo l'overflow: variabileImportante = %d\n", variabileImportante);

    return 0;
}
```

Fine della presentazione

Amedeo Natalizi