

Esercitazione W17D4

Buffer Overflow

Fabio Benevento - 28/02/2024

Traccia

Utilizzare il codice fornito per simulare un errore di buffer overflow.

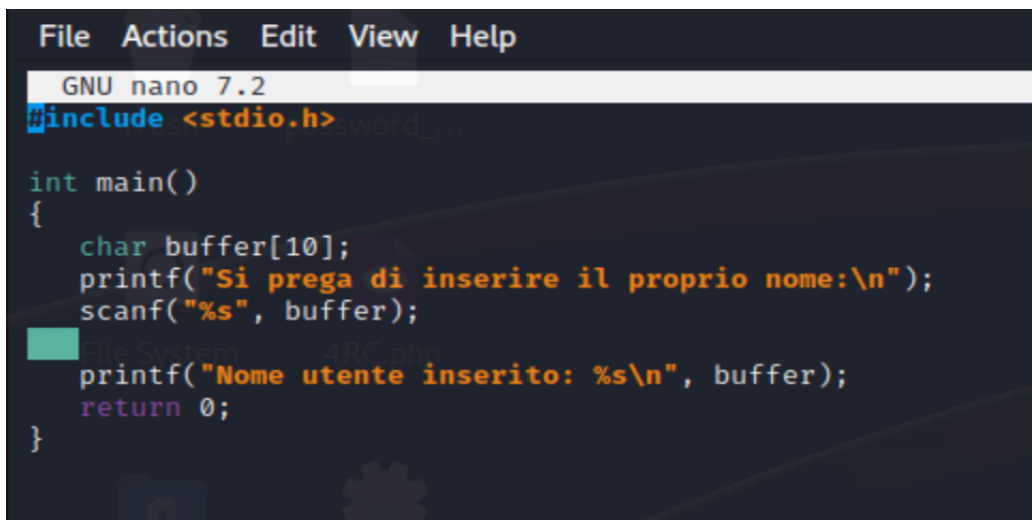
Il programma si avvia chiedendoci di inserire un nome utente:

- Inserendo un nome utente di 5 caratteri, il programma non ci riporta nessun problema, infatti come sappiamo il buffer accetta fino a 10 caratteri. Cosa succede se inseriamo 30 caratteri?
- Provate a riprodurre l'errore di segmentazione modificando il programma aumentando la dimensione del vettore a 30

Implementazione

Esercitazione guidata

Dopo aver aperto un editor di testo, nel caso specifico nano, ho editato il codice fornito a lezione come mostrato in figura.



```
File Actions Edit View Help
GNU nano 7.2
#include <stdio.h>

int main()
{
    char buffer[10];
    printf("Si prega di inserire il proprio nome:\n");
    scanf("%s", buffer);
    printf("Nome utente inserito: %s\n", buffer);
    return 0;
}
```

Ho compilato l'applicazione con il comando **gcc -g BOF.c -o BOF** ottenendo in output l'eseguibile **BOF**

```
(kali㉿kali)-[~/Desktop]
$ gcc -g BOF.c -o BOF
```

Ho lanciato l'applicazione con il comando **./BOF**.

Inserendo per il nome utente un numero di caratteri inferiore a 10 l'applicazione termina correttamente senza errori.

```
(kali㉿kali)-[~/Desktop]
$ ./BOF
Si prega di inserire il proprio nome:
aaaaaa
Nome utente inserito: aaaaaa
```

Nel caso invece si inseriscano in input un numero di caratteri superiore a 10, l'applicazione restituisce un errore di segmentation fault a segnalare il tentativo di accesso ad una zona di memoria non consentito.

```
(kali㉿kali)-[~/Desktop]
$ ./BOF
Si prega di inserire il proprio nome:
aaaaaaaaaaaaaaaaaaaaaaaaaaaaa
Nome utente inserito: aaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
zsh: segmentation fault ./BOF
```

Esercitazione autonoma

Il proseguo dell'esercitazione richiede di riprodurre lo stesso errore aumentando il numero di caratteri massimi consentiti a 30. Per fare ciò ho aumentato le dimensioni della variabile buffer in cui viene salvato il nome utente a 30 come mostrato in figura.

```
File Actions Edit View Help
GNU nano 7.2
#include <stdio.h>

int main()
{
    char buffer[30];
    printf("Si prega di inserire il proprio nome:\n");
    scanf("%s", buffer);
    printf("Nome utente inserito: %s\n", buffer);
    return 0;
}
```

Ho ricompilato l'applicazione con il comando **gcc -g BOF.c -o BOF**

```
(kali㉿kali)-[~/Desktop]
$ gcc -g BOF.c -o BOF
```

Ho lanciato l'applicazione con il comando **./BOF**.

Inserendo ora per il nome utente un numero di caratteri superiore a 10 caratteri (15 nel caso in esame) l'applicazione termina correttamente senza errori.

```
(kali㉿kali)-[~/Desktop]
$ ./BOF
Si prega di inserire il proprio nome:
aaaaaaaaaaaaaaaa
Nome utente inserito: aaaaaaaaaaaaaaaaaa
```

Nel caso invece si inseriscano in input un numero di caratteri superiore a 30, l'applicazione restituisce un errore di segmentation fault come in precedenza

```
(kali㉿kali)-[~/Desktop]
$ ./BOF
Si prega di inserire il proprio nome:
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
Nome utente inserito: aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
zsh: segmentation fault ./BOF
```

—