

Relazione sull'Esercizio di Buffer Overflow

Procedura

1. Navigazione alla Cartella di Lavoro

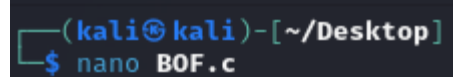
Abbiamo iniziato aprendo il terminale e navigando alla cartella Desktop del nostro ambiente Kali Linux con il comando:

```
cd /home/Kali/Desktop
```

2. Creazione del File Sorgente in C

Abbiamo creato un file chiamato BOF.c utilizzando un editor di testo (ad esempio nano, vim, o gedit). Il comando per creare e aprire il file con nano è:

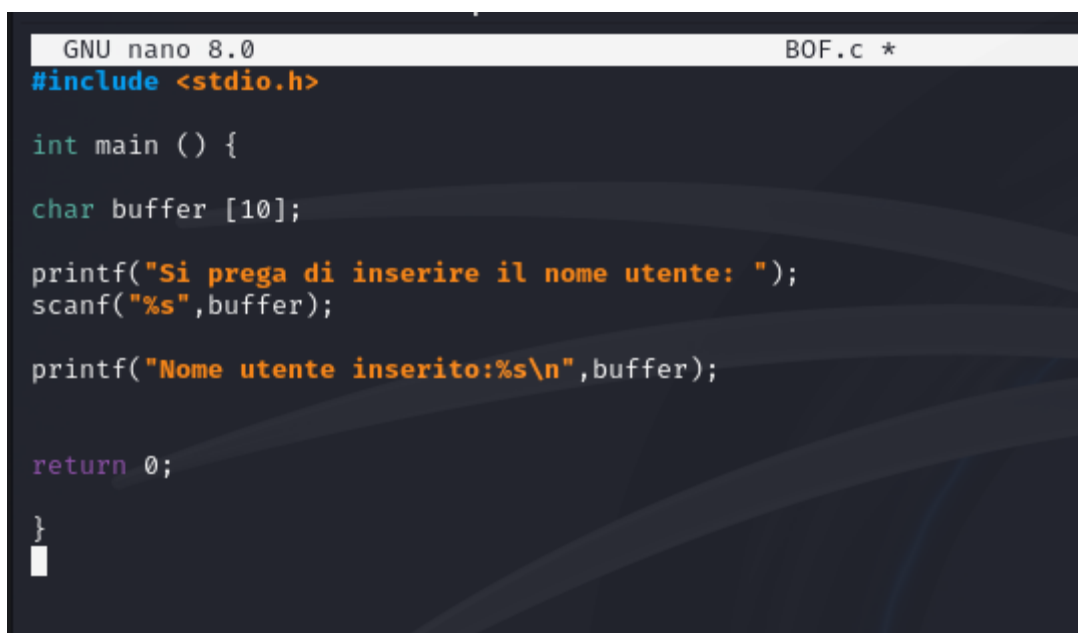
```
nano BOF.c
```



```
(kali@kali)-[~/Desktop]  
$ nano BOF.c
```

3. Scrittura del Codice

Nel file BOF.c abbiamo inserito il seguente codice C:



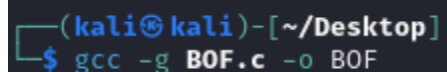
```
GNU nano 8.0 BOF.c *  
#include <stdio.h>  
  
int main () {  
    char buffer [10];  
  
    printf("Si prega di inserire il nome utente: ");  
    scanf("%s",buffer);  
  
    printf("Nome utente inserito:%s\n",buffer);  
  
    return 0;  
}
```

4. Compilazione del Programma

Abbiamo compilato il programma utilizzando il compilatore GCC con il comando:

```
gcc -o BOF BOF.c
```

Questo comando genera un eseguibile chiamato BOF.



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

5. Esecuzione del Programma

Abbiamo eseguito il programma con il comando:

```
./BOF
```

Test del Programma

1. Test Input Corto

Abbiamo inserito il nome utente "Ciccio" (6 caratteri) per verificare il funzionamento del programma:

Si prega di inserire il nome utente: Ciccio

Nome utente inserito: Ciccio

Il programma ha funzionato correttamente senza problemi, mostrando che il buffer può gestire un input inferiore a 10 caratteri.

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

2. Test con Input Lungo

Abbiamo inserito un nome utente con più di 20 caratteri per verificare la gestione del buffer overflow:

Si prega di inserire il nome utente: 123456789012345678901

```
(kali㉿kali)-[~/Desktop]
$ ./BOF
Si prega di inserire il nome utente: 123456789012345678901
Nome utente inserito:123456789012345678901
zsh: segmentation fault ./BOF
```

Il programma è andato in errore, generando un "segmentation fault" (errore di segmentazione). Questo errore si verifica perché l'input eccede la dimensione del buffer, sovrascrivendo la memoria adiacente e causando il crash del programma.

3. Correzione del Buffer Overflow

Per prevenire il buffer overflow, abbiamo modificato il codice aumentando la dimensione del buffer a 30 caratteri:

```
GNU nano 8.0 BOF.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;
}
```

Dopo aver aggiornato il codice, abbiamo ricompilato ed eseguito nuovamente il programma:

```
gcc -o BOF BOF.c
```

```
./BOF
```

Test del Programma Corretto

Abbiamo nuovamente inserito un nome utente con più di 20 caratteri:

```
Si prega di inserire il nome utente: 123456789012345678901
```

```
Nome utente inserito: 123456789012345678901
```

Il programma ha funzionato correttamente senza generare errori di segmentazione, dimostrando che il buffer di 30 caratteri può gestire input più lunghi.

```
(kali@kali)-[~/Desktop]
$ ./BOF
Si prega di inserire il nome utente: 123456789012345678901
Nome utente inserito:123456789012345678901
```