Consegna S7-L4

buffer overflow

INTRODUZIONE:

L'esercizio di oggi richiede di eseguire un buffer overflow scrivendo un codice appositamente errato per poi andarlo a correggere evitando che il problema si ripresenti.

COS’E´ UN BUFFER OVERFLOW (BOF):

Un buffer overflow è una vulnerabilità informatica che si verifica quando un programma, durante l'elaborazione di dati, supera il limite di memoria assegnato ad una zona di memoria temporanea (buffer) In pratica, un attaccante sfrutta questa vulnerabilità inserendo dati malevoli in un buffer, oltre il suo limite consentito. Questo può causare la sovrascrizione sulla memoria temporanea, danneggiando o sovrascrivendo dati importanti come istruzioni del programma stesso rendendo il sistema molto vulnerabile ad altri tipi di attacchi

come prima cosa ci spostiamo sul Desktop con il terminale tramite il comando "cd Desktop"(fig.1) così da poter creare un file .c chiamato "BOF.C" dove andremo a scrivere il codice, sbagliato, del programma; Per farlo usiamo il comando "nano BOF.C"(fig.2) una volta aperto andremo a scrivere il nostro codice(fig.3)

fig.1

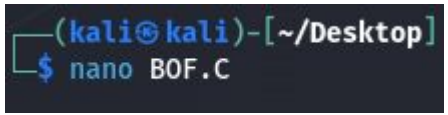

fig.2

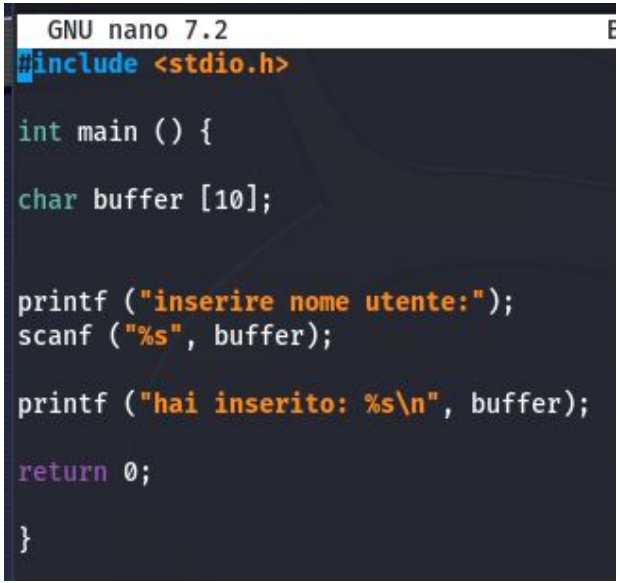

fig.3

il nostro prossimo step e' quello di assicurarci che il file sia stata creato(fig.4). Fatto ciò creiamo l'eseguibile tramite il comando "gcc -g BOF.C -o BOF"(fig.5), questo ci permetterà di avviare il programma, per farlo usiamo dunque il comando "./BOF"(fig.6a) il quale ci chiederà di inserire gli input e possiamo vedere che se restiamo nel limite scritto nel programma (ovvero di 10 caratteri) non succede nulla,vedi fig.6b, ma se oltrepassiamo quel numero il programma andrà automaticamente in errore vedi fig.7

fig.4

fig.5

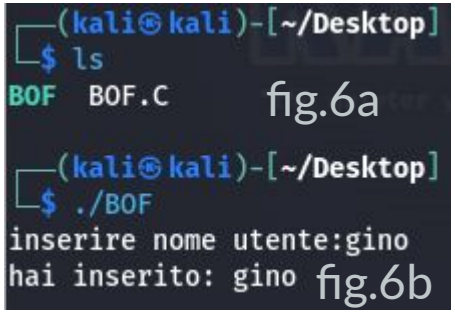

fig.6a

fig.6b

fig.7

```c
#include <stdio.h>

int main () {

char buffer [30];


printf ("inserire nome utente:");
scanf ("%30s", buffer);

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

return 0;

}
```

fig.8

fig.9

```
┌──(kali㉿kali)-[~/Desktop]
└─$ ./BOF
inserire nome utente:123456789101112131415161718119202122232425262728293031323334
hai inserito: 12345678910111213141516171811 9
```

infine per assicurarci che l'errore non si presenti più andiamo a correggere il codice semplicemente aggiungendo un valore più alto di caratteri e limitandolo nella funzione come possiamo vedere in "char buffer[30]"
e "scanf ("%30s", buffer)"(fig.8) dove prima erano "char buffer[10]" e "scanf ("%s", buffer)" . Per eseguire i cambiamente riutilizziamo il comando "gcc -g BOF.C -o BOF" così da aggiornare l'eseguibile e andiamo appunto ad eseguirlo e vediamo che i caratteri sono stati limitati e non abbiamo più l'errore(fig.9)