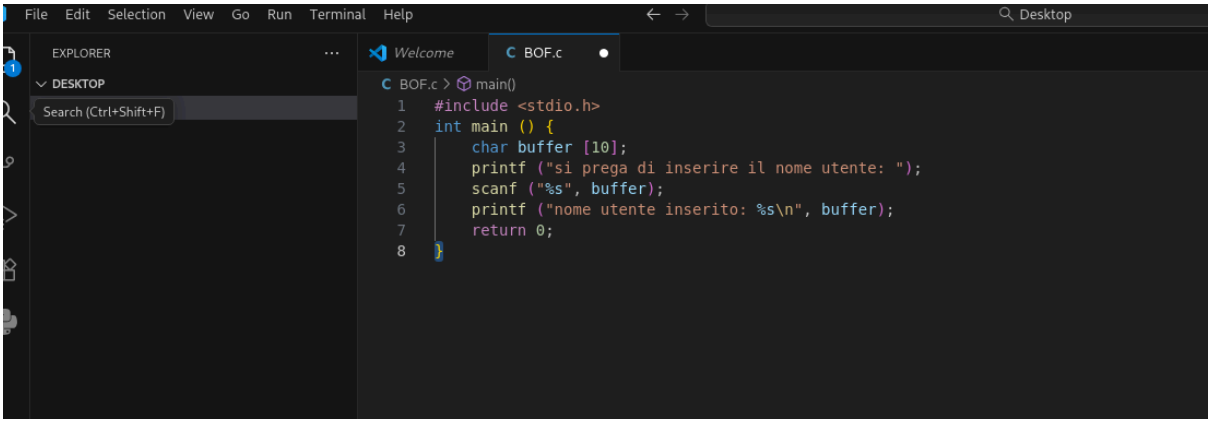
Traccia:

Il buffer overflow è una vulnerabilità conseguenza di una mancanza di controllo dei limiti dei buffer che accettano input utente.

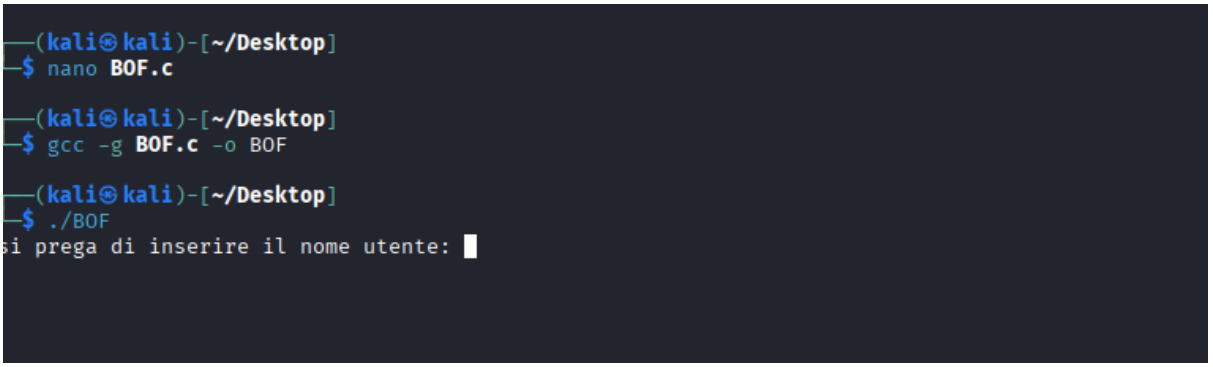
Nelle prossime slide vedremo un esempio di codice in C volutamente vulnerabile ai BOF, e come scatenare una situazione di errore particolare chiamata «segmentation fault», ovvero un errore di memoria che si presenta quando un programma cerca inavvertitamente di scrivere su una posizione di memoria dove non gli è permesso scrivere (come può essere ad esempio una posizione di memoria dedicata a funzioni del sistema operativo).

SVOLGIMENTO:



ho creato il codice come da slide per il BOF

con nano ho creato un documento BOF su desktop ho copiato il codice nel documento stando attenta a riportare il codice con ogni carattere per evitare errori, poi ho salvato con ctrl X e poi facendo y



```
┌──(kali㉿kali)-[~/Desktop]
└─$ ./BOF
si prega di inserire il nome utente: pippo
nome utente inserito: pippo

┌──(kali㉿kali)-[~/Desktop]
└─$
```

ho fatto partire il codice da terminale se rimango nei limiti del mio codice di BOF ovvero 10 caratteri mi accetta il nome utente

```
┌──(kali㉿kali)-[~/Desktop]
└─$ ./BOF
si prega di inserire il nome utente: fnjnsl4dnotndsontnctinolpdnikn
nome utente inserito: fnjnsl4dnotndsontnctinolpdnikn
zsh: segmentation fault  ./BOF

┌──(kali㉿kali)-[~/Desktop]
└─$
```

ma se ne inserisco di più (30 caratteri come da screenshot) il programma ci dà un messaggio di errore segmentation fault ovvero un errore che ti dice che stai cercando di scrivere in una porzione di memoria che non ti è permessa, il buffer in questo caso ha una memoria di massimo 10 caratteri quindi quello che è successo è che ho provato a scrivere più dati di quelli che stanno nel buffer
ma cosa succede se modifico il codice e il file aumentando la dimensione a 30?

BOF.c

```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;
}
```

```
Session  Actions  Edit  View  Help
  GNU nano 8.6
#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;
}
```

```
┌──(kali㉿kali)-[~/Desktop]
└─$ ./BOF
si prega di inserire il nome utente: gmnsklysnfopdhtdndyodjnspsjsklc
nome utente inserito: gmnsklysnfopdhtdndyodjnspsjsklc

┌──(kali㉿kali)-[~/Desktop]
└─$
```

ho aumentato la dimensione del vettore a 30 e sembra funzionare anche se ho inserito 31 caratteri tuttavia questo non basta ad eliminare la possibilità di BOF questo perchè io ho semplicemente aumentato lo spazio disponibile, non sto eliminando il problema, aumentare la dimensione del buffer riduce le possibilità di un BOF ma non elimina il rischio.

```
┌──(kali㉿kali)-[~/Desktop]
└─$ ./BOF
si prega di inserire il nome utente: hnfjsnsknfjsns78sjshnflsojspshnfyisnufnsjfimsnufhssduifhdsjfsfsfsjfghjsgfjshfgsmfgh
nome utente inserito: hnfjsnsknfjsns78sjshnflsojspshnfyisnufnsjfimsnufhssduifhdsjfsfsfsjfghjsgfjshfgsmfgh
zsh: segmentation fault  ./BOF
```

infatti aumentando di molto il numero dei caratteri mi ha dato di nuovo l’errore del segmentation fault.