

# Esercizio 09/11/2023

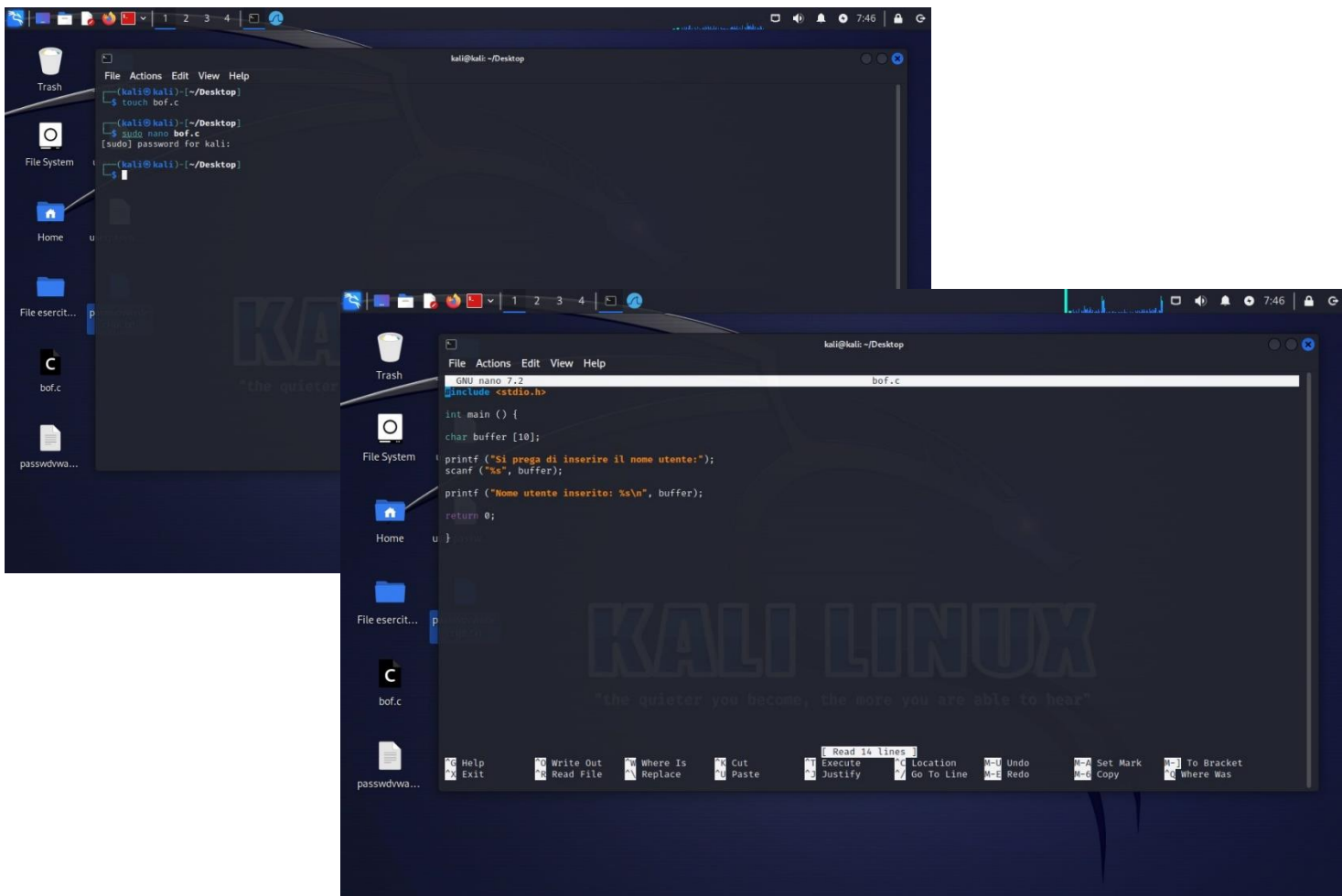
Nell'esercizio odierno ci viene proposto di testare un buffer overflow su un codice .c prima utilizzando il vettore preimpostato e poi modificandolo, ed eventualmente di risolvere i problemi che si presentano

## Buffer Overflow

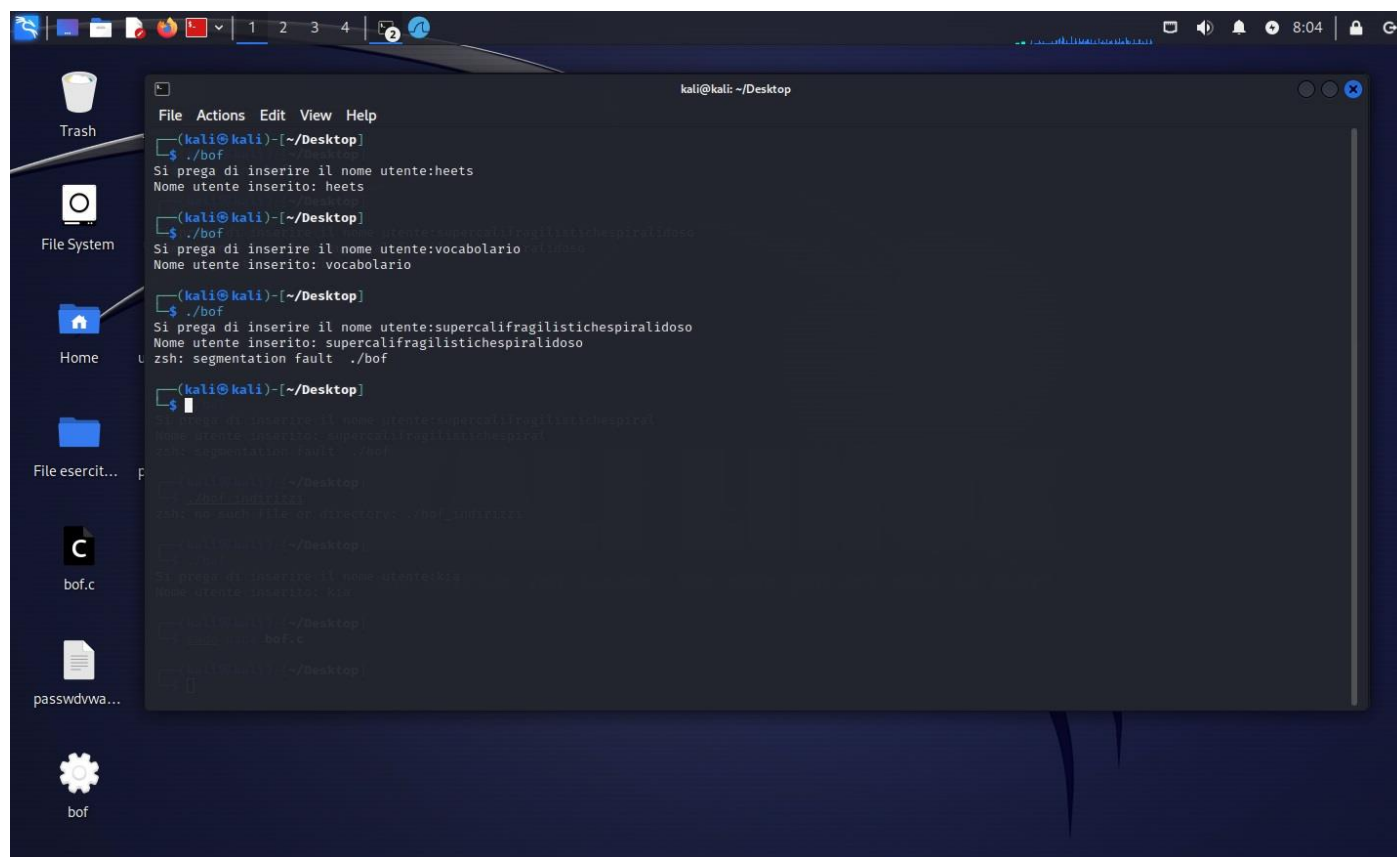
Il buffer overflow è una vulnerabilità attraverso la quale un attaccante sovraccarica la memoria buffer, va cioè ad immettere nella memoria buffer un numero di dati superiori alla memoria buffer stessa. Con questa vulnerabilità l'attaccante può controllare il flusso del programma, eseguire un codice malevolo o far crashare il programma stesso, o addirittura l'intero sistema operativo. Il BOF si verifica quando il programma scrive più dati di quelli che il buffer può gestire causando la sovrascrittura della memoria adiacente (quella immediatamente successiva o contigua al buffer interessato) e causando, dunque, comportamenti imprevisti, malfunzionamenti o portare a vulnerabilità di sicurezza.

**N.B.** Il buffer è uno spazio di memoria temporanea che contiene dati come input utente, parti di file o video e altro e hanno una dimensione finita, ossia possono contenere solo un certo quantitativo di dati. Immaginateli come una grandezza definita, un numero, cioè, che va ipoteticamente da 0 a 10 all'interno di parentesi quadre (quel numero definisce un vettore di interi). Ora immaginiamo che uno sviluppatore di un'applicazione non abbia dato limiti ai buffer, un attaccante potrà trovare un modo per scrivere dei dati oltre quei limiti.

Prima di tutto ho creato il documento con estensione .c



Verifica del funzionamento fino a 11 caratteri andata a buon fine, oltre ha dato ovviamente l'errore



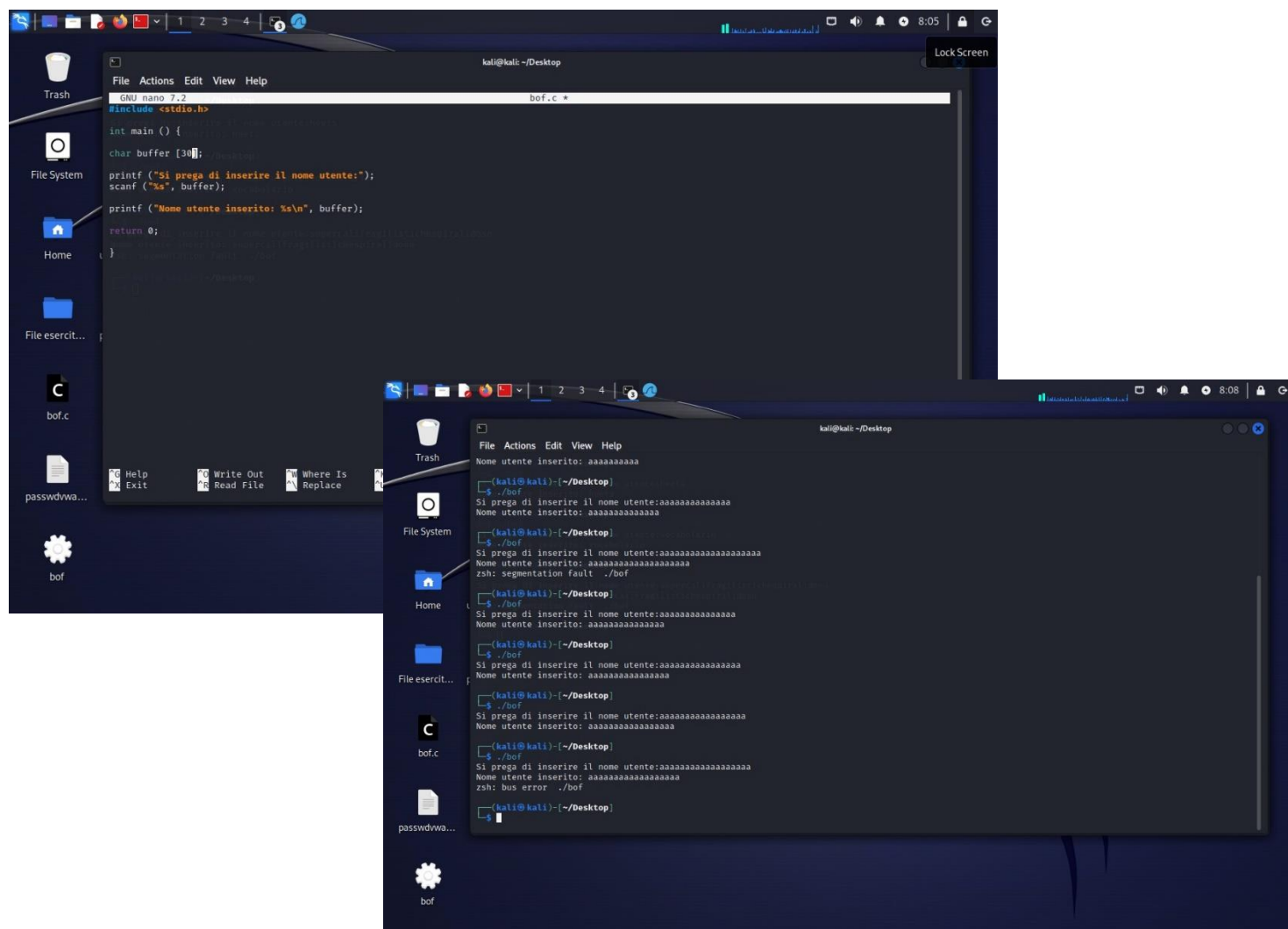
```
kali@kali: ~/Desktop
File Actions Edit View Help
(kali@kali)-[~/Desktop]
$ ./bof
Si prega di inserire il nome utente:heets
Nome utente inserito: heets

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

(kali@kali)-[~/Desktop]
$ ./bof
Si prega di inserire il nome utente:supercalifragilistichepsalidoso
Nome utente inserito: supercalifragilistichepsalidoso
zsh: segmentation fault ./bof

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

A questo punto ho modificato la dimensione del vettore e ho fatto una verifica che, comunque, non ha funzionato: fino a 17 li ha presi, a 18 ha dato il bus error e a 19 il segmentation fault



```
GNU nano 7.2 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;
}

(kali@kali)-[~/Desktop]
$ ./bof
Nome utente inserito: aaaaaaaaaa

(kali@kali)-[~/Desktop]
$ ./bof
Si prega di inserire il nome utente:aaaaaaaaaaaaaaaaaaaaa
Nome utente inserito: aaaaaaaaaaaaaaaaaaaaaa
zsh: segmentation fault ./bof

(kali@kali)-[~/Desktop]
$ ./bof
Si prega di inserire il nome utente:aaaaaaaaaaaaaaaaaaaaa
Nome utente inserito: aaaaaaaaaaaaaaaaaaaaaa

(kali@kali)-[~/Desktop]
$ ./bof
Si prega di inserire il nome utente:aaaaaaaaaaaaaaaaaaaaa
Nome utente inserito: aaaaaaaaaaaaaaaaaaaaaa

(kali@kali)-[~/Desktop]
$ ./bof
Si prega di inserire il nome utente:aaaaaaaaaaaaaaaaaaaaa
Nome utente inserito: aaaaaaaaaaaaaaaaaaaaaa
zsh: bus error ./bof

(kali@kali)-[~/Desktop]
$ ./bof
Si prega di inserire il nome utente:aaaaaaaaaaaaaaaaaaaaa
Nome utente inserito: aaaaaaaaaaaaaaaaaaaaaa
zsh: segmentation fault ./bof
```

Quindi sono andata a modificare la variabile % numerandola, in questo modo mi ha permesso di arrivare fino a 29 caratteri e, come possiamo vedere dalla seconda immagine di verifica, oltre i 29 caratteri viene accettato comunque il nome ma lo visualizza solo fino al ventinovesimo carattere

