M4-Consegna week 17 day 4 – Daniele Rufo

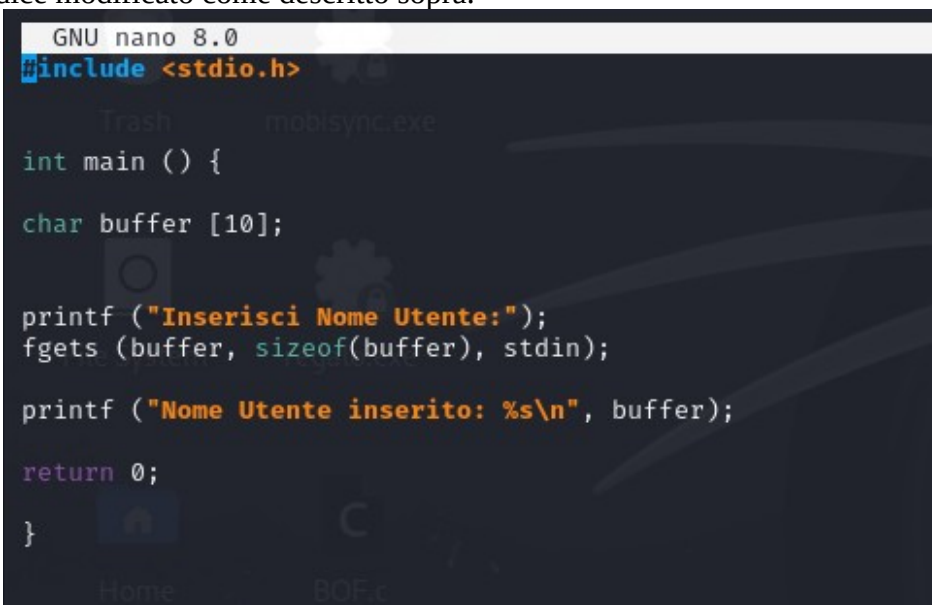
BUFFER OVERFLOW

L'esercizio chiede di mettere mano ad un codice in C scritto per causare il buffer overflow. Avendo una conoscenza limitata di C ho cercato soluzioni semplici, di facile comprensione ma che mi dessero una chiara idea della radice del problema e delle loro azioni.
Prima ho verificato quanto riportato nelle slide, cioè ho copiato il codice dato da epicode e verifaco che desse problemi di BOF (buffer overflow).

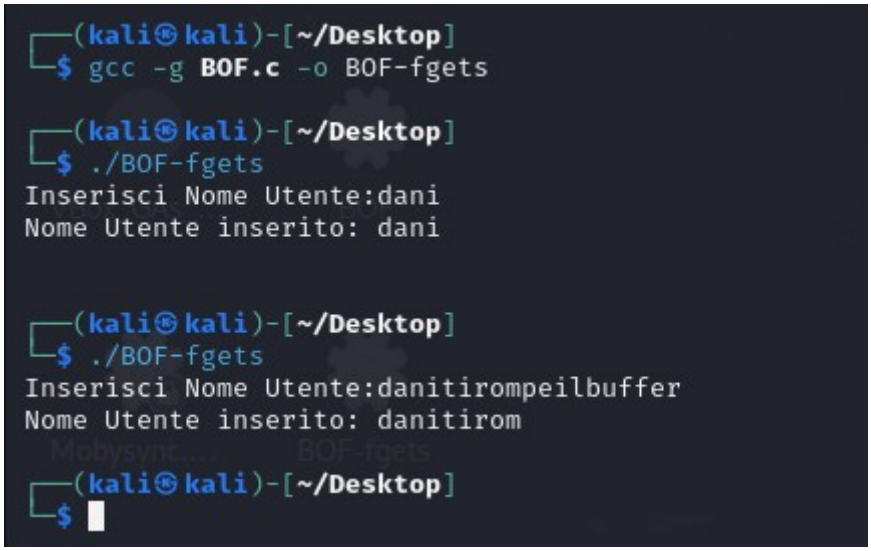
Una volta fatto ciò ho cercato possibili soluzioni. Ne ho trovate tre più o meno semplici e ne ho scelta una da implementare. Le opzioni sono:

- usare delle funzioni differenti da per leggere l'input utente e verificare la dimensione del buffer
- inserire un ciclo che controlla la lunghezza del codice e procedere solo se è minure di quella del buffer
- allocazione dinamica di memoria in base alla lunghezza dei dati. Questa l'ho trovata più difficile da implementare e nonostante sia una valida opzione non sono riuscito a scrivere un codice che funzionasse

Ho scelto la prima opzione perché mi sembrava comprensibile e immediata poiché usa una funzione *fgets* () che ritorna solamente ciò che non eccede la lunghezza del buffer -*sizeof(buffer)*-.
Ecco il codice modificato come descritto sopra.



Qui invece riporto la verifica di un output utente che non rispetta le dimensioni del buffer.



Se osserviamo il comportamento del programma vediamo che riporta in output solamente i caratteri che riempiono il buffer fino al massimo della sua capienza e non oltre. Ciò impedisce il buffer overflow.

Grazie