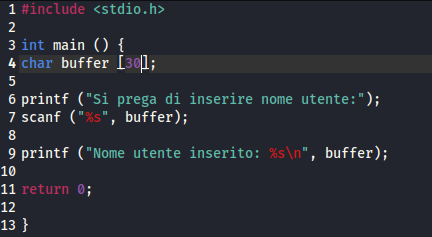
S7/L4

Traccia:

Scatenare una situazione di errore da un codice in C 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.

Per prima cosa creiamo un documento su Kali con l’estensione .c e andiamo a trascrivere il seguente programma in codice C:



Vediamo brevemente il codice:

#include <stdio.h> necessario per utilizzare le funzioni di input/output (printf e scanf) standard in C.

Int main (); è la funzione principale del programma ed “int” indica che il programma restituirà un valore intero.

Char buffer [30]; consiste in un array di caratteri chiamato buffer con dimensione 30 e sarà utilizzato per memorizzare il nome utente inserito.

Printf; stampa il messaggio come output, in questo caso si chiede di inserire un nome utente.

Scanf(“%s”, buffer); legge una stringa di caratteri inserita dall'utente. Printf(“nome utente inserito: %s\n”, buffer; stampa il nome utente inserito.

Return 0; Indica che il programma è terminato correttamente.

In generale il programma richiede all'utente di inserire “un nome utente” e di stampare il nome inserito come output. Tuttavia se usiamo “%s” potremmo incorrere in un problema se l'utente inserisce più di 30 caratteri, questo perché “scanf” tenterebbe di memorizzare tutti i caratteri in buffer, causando un overflow e comportamenti indefiniti nel programma.

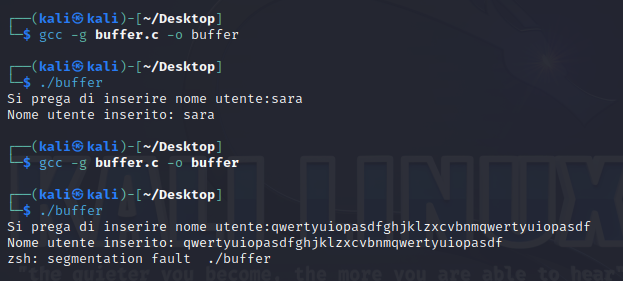
Potremmo per esempio usare “%9s”, in modo tale che scanf leggerà al massimo 30 caratteri dalla stringa inserita dall'utente, prevenendo così un potenziale overflow e garantendo che ci sia spazio sufficiente.

Ma che cos’è un buffer overflow?

È una vulnerabilità informatica che si verifica quando vengono scritti dei dati che vanno oltre i limiti di un buffer di memoria (ovvero una zona di memoria volatile contenente dati temporanei).

Se questo input non viene sanato dal programmatore e quindi i dati inseriti dall’utente superano i limiti dell’effettiva portata di archiviazione del buffer stesso, un attaccante può sfruttarlo come vettore di attacco. Essenzialmente si possono inviare enormi quantità di dati portando ad uno straripamento della memoria che porta poi il programma a crashare, oppure la parte di dati in eccesso può far parte di un codice malevolo. Inviando poco a poco pezzi di codice malevolo si arriva alla sua integrità, ed infine l’utente si ritroverà un malware. Per arrivare a ciò in una sola volta si possono usare i puntatori ovvero variabili che puntano a una cella di memoria specifica, iniettando il malware.

Fatto ciò apriamo un terminale su Kali e andiamo a scrivere i seguenti comandi:



Per cui eseguiamo il programma con il comando ./ (seguito dal nome del documento).

Il programma si avvia chiedendo di inserire un nome utente; per prima cosa proviamo con “Sara” ovvero una stringa di 4 caratteri e come vediamo il programma lo accetta; questo perché nel codice abbiamo inserito che il buffer può accettare fino a [30] caratteri.

Infatti, come possiamo vedere se inseriamo un nome utente più grande di 30 caratteri, ci tornerà la scritta “segmentation fault”, ovvero errore di segmentazione.

L’errore di segmentazione avviene quando un programma, tenta di scrivere contenuti su una porzione di memoria alla quale non ha accesso, di conseguenza alcuni caratteri stanno sovrascrivendo aree di memorie inaccessibili.