BUFFER OVERFLOW:
SCRIPT IN C DA #KALI LINUX

TRACCIA:

NELLA LEZIONE DEDICATA AGLI ATTACCHI DI SISTEMA, ABBIAMO PARLATO DEI **BUFFER OVERFLOW**, UNA VULNERABILITÀ CHE È 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).

RIPORTATE IL CODICE CHE SEGUE SULLA VOSTRA KALI LINUX, CREANDO UN NUOVO DOCUMENTO CON ESTENSIONE **.C** SUL DESKTOP:

```
#include <stdio.h>
int main () {
    char buffer [10];
printf ("Si prega di inserire il nome utente:");
scanf ("%s", buffer);

printf ("Nome utente inserito: %s\n", buffer);
return 0;
}
```

PER CREARE UN NUOVO DOCUMENTO SU KALI, AVVIATE LA VOSTRA KALI LINUX, ED UNA VOLTA PRESENTATA LA SCHERMATA PRINCIPALE CLICCATE SULL'ICONA DEL TERMINALE. DAL TERMINALE, POI, SPOSTATEVI SUL DESKTOP ESEGUENDO IL COMANDO:
cd /home/Kali/Desktop OPPURE **ls -> cd Desktop**
SUCCESSIVAMENTE, ESEGUIAMO **L'EDITOR DI TESTO NANO**, CHE CI PERMETTE O DI APRIRE UN FILE ESISTENTE OPPURE DI CREARNE UNO NUOVO SE IL NOME DEL FILE SPECIFICATO NON ESISTE. ESEGUIAMO QUINDI DAL TERMINALE IL COMANDO DI SEGUITO PER CREARE UN FILE BOF.C
sudo nano BOF.C
RIPORTATE IL FRAMMENTO DI CODICE CHE AVETE APPENA CREATO, FACENDO ATTENZIONE A RIPORTARE TUTTI I SIMBOLI, POTETE MODIFICARE A VOSTRO PIACIMENTO IL CONTENUTO DELLE «PRINTF» TRA LE VIRGOLETTE.
UNA VOLTA COMPLETATO, CHIUDETE E SALVATE IL FILE. PER CHIUDERE E SALVARE UN FILE CON L'EDITOR DI TESTO NANO, DOVETE SEGUIRE UNA SEQUENZA DI COMANDI DA TASTIERA:
1. PREMETE INSIEME I TASTI CTRL E X SULLA VOSTRA TASTIERA
2. IL TOOL VI CHIEDERÀ SE VOLETE SALVARE IL PROGRAMMA. DIGITATE «Y» E POI PREMETE «INVIO» PER SALVARE IL FILE CON IL NOME SCELTO

A QUESTO PUNTO, COMPILATE IL FILE UTILIZZANDO IL COMANDO
gcc –g BOF.C –o BOF
RICORDATE DI COMPILARE IL PROGRAMMA AD OGNI MODIFICA DEL CODICE. UNA VOLTA FATTO, POTETE ESEGUIRE IL PROGRAMMA ESEGUENDO IL COMANDO **./BOF**

IL PROGRAMMA SI AVVIA CHIEDENDOCI DI INSERIRE UN NOME UTENTE:
- INSERENDO UN NOME UTENTE DI **5 CARATTERI**, IL PROGRAMMA NON CI RIPORTA **NESSUN PROBLEMA**, INFATTI COME SAPPIAMO **IL BUFFER ACCETTA FINO A 10 CARATTERI**. COSA SUCCEDE SE INSERIAMO 30 CARATTERI? PROVIAMO.
- SE INSERIAMO 30 CARATTERI IL PROGRAMMA CI RITORNA UN ERRORE, «SEGMENTATION FAULT», OVVERO ERRORE DI SEGMENTAZIONE. L'ERRORE DI SEGMENTAZIONE AVVIENE QUANDO UN PROGRAMMA, COME ABBIAMO DETTO IN PRECEDENZA, TENTA DI SCRIVERE CONTENUTI SU UNA PORZIONE DI MEMORIA ALLA QUALE NON HA ACCESSO.

QUESTO È UN CHIARO ESEMPIO DI BOF, ABBIAMO INSERITO 30 CARATTERI IN UN BUFFER CHE NE PUÒ CONTENERE SOLAMENTE 10 E DI CONSEGUENZA ALCUNI CARATTERI STANNO SOVRASCRIVENDO AREE DI MEMORIE INACCESSIBILI.

PROVATE A RIPRODURRE L'ERRORE DI SEGMENTAZIONE MODIFICANDO IL PROGRAMMA COME DI SEGUITO:
- AUMENTANDO LA DIMENSIONE DEL VETTORE A 30

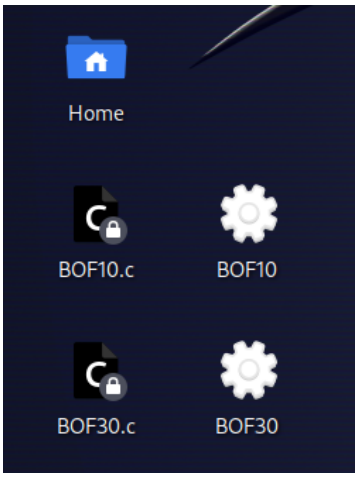
./BOF10:

BUFFER OVERFLOW -> IMMAGAZZINA PIÙ MEMORIA DI QUANTA NE POSSA ESSERE CONTENUTA, COSÌ LE INFORMAZIONI CHE INSERISCO SATURANO QUESTO BUFFER E VEGONO SCRITTE NELLA RAM AL DI FUORI DI CIÒ CHE È CONSENTITO. QUESTA VULNERABILITÀ AVVIENE PERCHÈ NON CI SONO CONTROLLI ADATTI SUI LIMITI DEL BUFFER E SUGLI INPUT DELL'UTENTE.

NON SEMPRE UN BOF FINISCE IN UN "SEGMENTATION FAULT", MA QUESTO ACCADE QUANDO IL CONTENUTO ARRIVA AD INTACCARE ANCHE AREE DI MEMORIA IN CUI A QUESTO PROCESSO NON È CONSENTITA LA SCRITTURA, CIOÈ QUANDO UN PROGRAMMA CERCA DI SCRIVERE SU UNA PARTE DI MEMORIA DOVE NON GLI È PERMESSO SCRIVERE AD ESEMPIO UN'AREA DI MEMORIA DI ALTRI PROCESSI, O UN'AREA DI MEMORIA DEDICATA AL SISTEMA OPERATIVO... QUANDO UN PROCESSO VIENE AVVIATO, UN OS SI ATTIVA AD ALLOCARE UN'AREA DI MEMORIA DEDICATA A QUESTO PROCESSO, QUINDI SI VA IN "SEGMENTATION FAULT" QUANDO C'È UNO SCAVALCAMENTO DELL'AREA CONSENTITA.

PROVO A PROCEDERE CON L'ESERCIZIO DEDICANDO 10 BYTE, CIOÈ 10 CARATTERI, AL BUFFER. UNA VOLTA COMPILATO CHIAMERÒ QUESTO FILE DI CODICE BOF10

E SUCESSIVAMENTE AUMENTO IL BUFFER A 30 BYTE, CIOÈ 30 CARATTERI ED UNA VOLTA COMPILATO CHIAMERÒ QUESTO FILE DI CODICE BOF30. LI SALVO ENTRAMBI SUL DESKTOP.




```
┌──(kali㉿kali)-[~]
└─$ cd Desktop

┌──(kali㉿kali)-[~/Desktop]
└─$ ./BOF10
Inserisci il nome utente:AAAAABBBBBCCCCCCCC
Nome utente iserito: AAAAABBBBBCCCCCCCC
zsh: bus error  ./BOF10

┌──(kali㉿kali)-[~/Desktop]
└─$ ./BOF10
Inserisci il nome utente:AAAAABBBBBCCCCCCCCD
Nome utente iserito: AAAAABBBBBCCCCCCCCD
zsh: segmentation fault  ./BOF10
```

```
┌──(kali㉿kali)-[~/Desktop]
└─$ ./BOF30
Inserisci il nome utente:AAAAAAAAAABBBBBBBBBBCCCCCCCCCCDDDDDDDDE
Nome utente iserito: AAAAAAAAAABBBBBBBBBBCCCCCCCCCCDDDDDDDDE

┌──(kali㉿kali)-[~/Desktop]
└─$ ./BOF30
Inserisci il nome utente:1111111111222222222233333333344444444455
Nome utente iserito: 111111111122222222223333333334444444455
zsh: bus error  ./BOF30

┌──(kali㉿kali)-[~/Desktop]
└─$ ./BOF30
Inserisci il nome utente:11111111112222222222333333333444444445556
Nome utente iserito: 1111111111222222222233333333344444444556
zsh: segmentation fault  ./BOF30
```

POSSO NOTARE COME SATURANDO I CARATTERI A DOSPOSIZIONE AL MINIMO NECESSARIO (10 NEL PRIMO CASO E 30 NEL SECONDO), NON SI VA SUBITO IN SEGMENTATION FAULT, PERCHÈ L'INDIRIZZO CHE DEVO ANDARE A SOVRASCRIVERE È NELL'OTTETTO DI MEMORIA SUCCESSIVO. AGGIUNGENDO QUINDI ALTRI 8 CARATTERI +1 PER SUPERARLO ESCO DAL BUS ERROR, E RAGGIUNGO IL SEGMENTATION FAULT.