

BUFFER OVERFLOW

Nell'esercizio di oggi andremo ad eseguire un attacco di buffer overflow, una vulnerabilità che è conseguenza di una mancanza di controllo dei limiti dei buffer che accettano input utente.

Andremo a scrivere un codice in linguaggio C volutamente vulnerabile ai BOF ed a 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.

Andremo quindi a spostarci nella directory "Desktop" per scrivere il nostro file.c .

```
File Actions Edit View Help
(kali@kali)-[~]
$ cd /home/kali/Desktop
(kali@kali)-[~/Desktop]
$ nano BOF.c
```

```
kali@kali: ~/Desktop
File Actions Edit View Help
GNU nano 6.4 BOF.c
#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;

}
```

A questo punto andremo a compilare il file tramite il comando `gcc -g BOF.c -o BOF`.

```
File Actions Edit View Help
(kali㉿kali)-[~]
$ cd /home/kali/Desktop
(kali㉿kali)-[~/Desktop]
$ nano BOF.c
(kali㉿kali)-[~/Desktop]
$ gcc -g BOF.c -o BOF
```

Andremo così ad eseguire il programma con il comando ./BOF

```
(kali㉿kali)-[~/Desktop]
$ gcc -g BOF.c -o BOF
(kali㉿kali)-[~/Desktop]
$ ./BOF
si prega di inserire il nome utente: 
```

Il programma si avvierà chiedendoci di inserire il nome utente, come prima cosa inseriremo un nome utente di pochi caratteri.

```
(kali㉿kali)-[~/Desktop]
$ ./BOF
si prega di inserire il nome utente: test1
nome utente inserito: test1
(kali㉿kali)-[~/Desktop]
$ 
```

Come vediamo il programma non riporta nessun problema, in quanto il buffer accetta fino a 10 caratteri. Proveremo ora ad inserire 30 caratteri.

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

Il programma ritornerà un errore “segmentation fault”, ovvero errore di segmentazione. Esso avviene quando un programma tenta di scrivere contenuti su una porzione di memoria alla quale non ha accesso; questo è un esempio di buffer overflow.

Andremo ora a modificare la dimensione del vettore del buffer a 30.

```
kali@kali: ~/Desktop
File Actions Edit View Help
GNU nano 6.4 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;

}
```

Andremo ora ad inserire un nome utente più lungo verificando che non ci sia un errore di buffer.

```
(kali@kali)-[~/Desktop]
$ ./BOF
si prega di inserire il nome utente: fhfhdsjasohsohwddiwdvwhdewf
nome utente inserito: fhfhdsjasohsohwddiwdvwhdewf
zsh: segmentation fault ./BOF

(kali@kali)-[~/Desktop]
$
```