REPORT 08/12/2022

FACCILONGO DOMENICO

BUFFER OVERFLOW

TASK

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).

Provate a riprodurre l'errore di segmentazione modificando il programma come di seguito:

- Aumentando la dimensione del vettore a 30;

ANALISI E VALUTAZIONE

Per prima cosa creiamo un file.c e compiliamo il codice seguente:

```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;
}
```

Successivamente inseriamo il nome utente che ci chiede, e in effetti il programma non presenta nessun problema, ma se provassimo ad inserire 30 caratteri ci ritornerà un errore di "segmentation fault".

```
┌──(kali㉿kali)-[~/Desktop]
└─$ gcc -g BOF.c -o BOF

┌──(kali㉿kali)-[~/Desktop]
└─$ ./BOF
Si prega di inserire il nome utente: test1
Nome utente inserito: test1

┌──(kali㉿kali)-[~/Desktop]
└─$ ./BOF
Si prega di inserire il nome utente: dsadasndhsdyhsdndhejdfnfjhefnef
Nome utente inserito: dsadasndhsdyhsdndhejdfnfjhefnef
zsh: segmentation fault  ./BOF
```

Questo è un esempio di Buffer overflow in cui abbiamo inserito 30 caratteri in un buffer che ne può contenere solamente 10 e di conseguenza alcuni caratteri stanno sovrascrivendo aree di memoria inaccessibili.

Andiamo a modificare il codice per poter inserire aree di memoria accessibili per un tot. Di 30 caratteri:

```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;
}
```

Facciamo un test per vedere se è verificato:

```
┌──(kali㉿kali)-[~/Desktop]
└─$ ./BOF
Si prega di inserire il nome utente: test1234567890
Nome utente inserito: test1234567890

┌──(kali㉿kali)-[~/Desktop]
└─$ ./BOF
Si prega di inserire il nome utente: dasdasdasdasdasdasdasdas
Nome utente inserito: dasdasdasdasdasdasdasdas
```