

Report Esercizio

W17-D5



Redatto da Andrea Sciattella

18/06/2024

Errore. Per applicare Title al testo da visualizzare in questo punto, utilizzare la scheda Home.

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).

Errore. Per applicare Title al testo da visualizzare in questo punto, utilizzare la scheda Home.

SVOLGIMENTO ESERCIZIO

- Accediamo al desktop della macchina kali, e creiamo il documento intitolato “BoF.c” con il comando touch

```
File Actions Edit View Help
File Edit Options Buffers Tools C Help
#include <stdio.h>
#include <string.h>

void vulnerableFunction(char *str) {
    char buffer[16]; // Un piccolo buffer di 16 byte
    strcpy(buffer, str); // Copia la stringa di input nel buffer senza controllo
    printf("Buffer content: %s\n", buffer);
}

int main(int argc, char *argv[]) {
    if (argc != 2) {
        printf("Usage: %s <input string>\n", argv[0]);
        return 1;
    }
    vulnerableFunction(argv[1]);
    return 0;
}
```

- Ora compiliamo il file con estensione C rinominandolo in “BoF”, tramite il compiler GCC

```
(kali@kali)-[~/Desktop]
$ gcc BoF.c -o Bof
```

- Run the program e vediamo se il programma ha funzionato

```
(kali@kali)-[~/Desktop]
$ ./BoF AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
Buffer content: AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
zsh: segmentation fault ./BoF AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
```

- La frase “zsh: segmentation fault ./BoF AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA” conferma quindi il funzionamento del programma, quindi un Buffer Overflow.