Progetto S7-L4

Studente: Simone Mininni

Task: Verificare l ‘errore di buffer overflow ed implementare soluzioni per controllare l ‘input utente.

Il buffer overflow si verifica ad esempio quando l‘input utente non controllato supera il limite della memoria ad esso destinato, ovvero inserisce più caratteri rispetto alla capienza massima definita dal programmatore per contenere la stringa e, quindi, vengono sovrascritti in locazioni di memoria adiacenti, non a loro destinate.
Questo potrebbe essere utilizzato dai black hat per controllare il flusso del programma e caricare in memoria un codice malevolo.

Codice di partenza:

```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;

}
```

Test legit:

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

Segmentation fault:

```
┌──(simone㉿kali)-[~/Desktop/BOF]
└─$ ./bof
Si prega di inserire il nome utente:1234567891011121314
Nome utente inserito: 1234567891011121314
zsh: segmentation fault  ./bof
```

Implementiamo la funzione fgets per controllare l' input dell' utente:

```c
#include <stdio.h>

int main(){


    char buffer[30];

    printf("Si prega di inserire il nome utente:");
    fgets(buffer, sizeof buffer, stdin);

    printf("Nome utente inserito: %s\n", buffer);

    return 0;

}
```

Check:

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

Anche immettendo più caratteri del limite dell' array, non si genera un errore di segmentation fault.