

EPICODE-CS0124

S7/L4 - Pratica

Flaviano Sedici

Indice

Traccia	3
1. Compilazione ed esecuzione del programma in C	4
2. Compilazione ed esecuzione del programma in C con 30 caratteri ...	5

Riferimenti e versioni

Responsabile/referente del documento: Flaviano Sedici

Versionamento

Versione	Descrizione	Ruolo	Data
1.0	Redazione documento	Responsabile	07/03/2024

Traccia

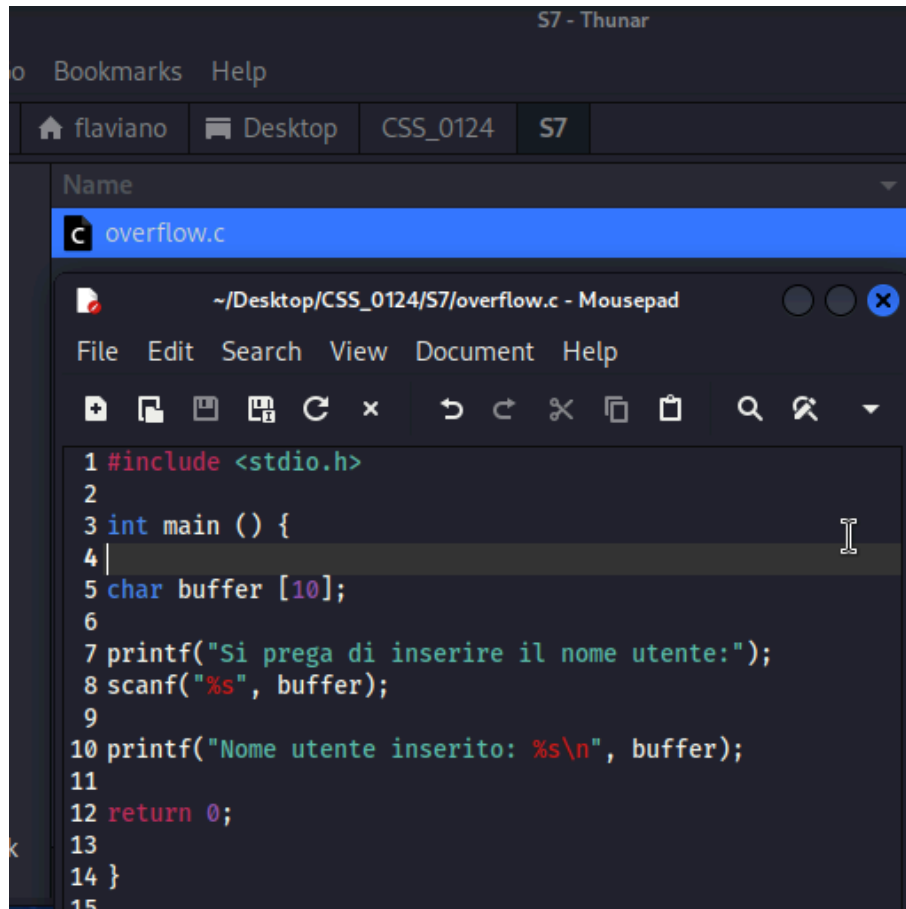
Hacking MS08-067

Abbiamo già parlato del 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).

1. Compilazione ed esecuzione del programma in C

Come indicato nella traccia abbiamo redatto e compilato il programma riportato nell'immagine.

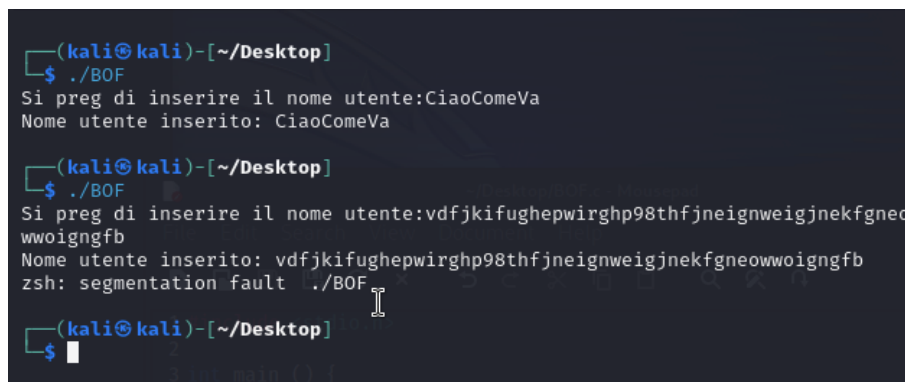


```

1 #include <stdio.h>
2
3 int main () {
4
5 char buffer [10];
6
7 printf("Si prega di inserire il nome utente:");
8 scanf("%s", buffer);
9
10 printf("Nome utente inserito: %s\n", buffer);
11
12 return 0;
13
14 }
15

```

Programma in C



```

(kali@kali)-[~/Desktop]
$ ./BOF
Si preg di inserire il nome utente:CiaoComeVa
Nome utente inserito: CiaoComeVa

(kali@kali)-[~/Desktop]
$ ./BOF
Si preg di inserire il nome utente:vdfjkifughepwirghp98thfjneignweigjnekfgneowwoigngfb
Nome utente inserito: vdfjkifughepwirghp98thfjneignweigjnekfgneowwoigngfb
zsh: segmentation fault ./BOF

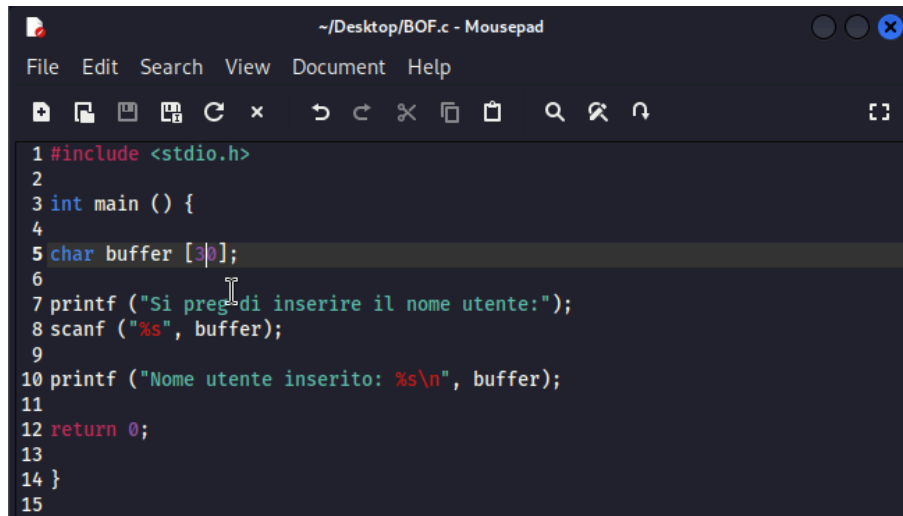
```

Esecuzione del programma

Come si evince dall'immagine inserendo 10 caratteri il programma viene eseguito normalmente, ma inserendo più caratteri il sistema restituisce un errore di **segmentation fault**.

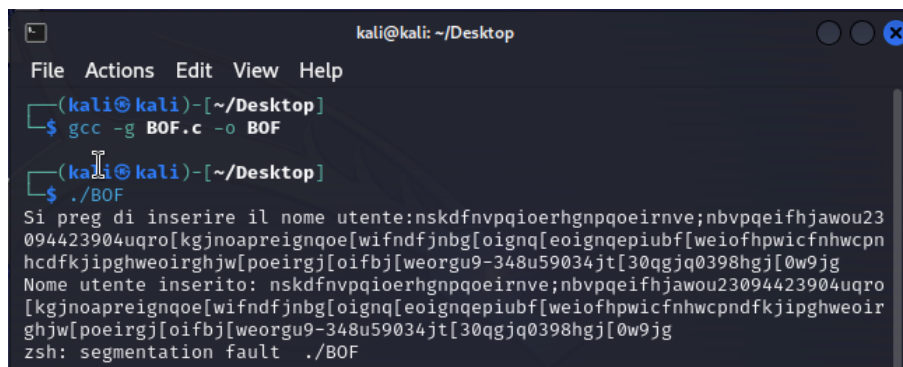
2. Compilazione ed esecuzione del programma in C con 30 caratteri

Abbiamo inserito come richiesto dalla traccia 30 caratteri nel buffer, abbiamo ripetuto la compilazione e l'esecuzione del programma.

A screenshot of a text editor window titled '~/.Desktop/BOF.c - Mousepad'. The window contains a C program with 15 lines of code. The code defines a character array 'buffer' of size 30 and prompts the user to enter a name. The program prints the entered name and then returns 0.

```
1 #include <stdio.h>
2
3 int main () {
4
5 char buffer [30];
6
7 printf ("Si prega di inserire il nome utente:");
8 scanf ("%s", buffer);
9
10 printf ("Nome utente inserito: %s\n", buffer);
11
12 return 0;
13
14 }
15
```

Programma in C

A screenshot of a terminal window titled 'kali@kali: ~/.Desktop'. It shows the compilation of the BOF.c program using gcc and its execution. The execution results in a segmentation fault due to a buffer overflow.

```
kali@kali: ~/.Desktop
File Actions Edit View Help
└─(kali@kali)-[~/Desktop]
└─$ gcc -g BOF.c -o BOF
└─(kali@kali)-[~/Desktop]
└─$ ./BOF
Si prega di inserire il nome utente:nskdfnvpqioerhgnpqoeirve;nbvpqeifhjawou23
094423904uqro[kgjnoapreignqoe[wifndfjnbgoignq[oeignqepiubf[weiofhpwicfnhwcpcn
hcdkfjipghweoirghjw[poeirgj[oifbj[weorgu9-348u59034jt[30qgj0398hgj[0w9jg
Nome utente inserito: nskdfnvpqioerhgnpqoeirve;nbvpqeifhjawou23094423904uqro
[kgjnoapreignqoe[wifndfjnbgoignq[oeignqepiubf[weiofhpwicfnhwcpcndfkjipghweoir
ghjw[poeirgj[oifbj[weorgu9-348u59034jt[30qgj0398hgj[0w9jg
zsh: segmentation fault ./BOF
```

Esecuzione del programma

Come si evince dall'immagine inserendo più di 30 caratteri il sistema restituisce un errore di **segmentation fault**.