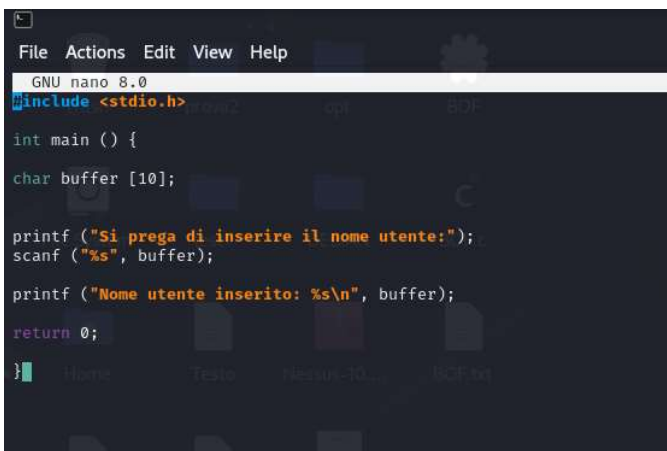


S7_L4

Traccia: Abbiamo già parlato del buffer overflow, una vulnerabilità che è conseguenza di una mancanza di controllo dei limiti dei buffer che accettano input utente.

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. Creiamo un file col nome BOF.c scrivendo un semplice script in linguaggio C nella quale il programmino richiede di inserire il nome utente



```
GNU nano 8.0
#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;
}
```

2. Abbiamo inserito da terminale il comando gcc per effettuare una compilazione dello script chiamato BOF nel file BOF.c creato



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

3. Abbiamo lanciato il programma con il comando ./ seguito dal nome del file BOF. Inserendo la parola “prova1” che è formata da 6 caratteri, non si riscontra nessun errore perché il vettore configurato è impostato per l’inserimento di un massimo di 10 caratteri



```
(kali@kali)-[~/Desktop]
$ nano BOF.c
(kali@kali)-[~/Desktop]
$ ./BOF
Si prega di inserire il nome utente:prova1
Nome utente inserito: prova1
```

- Provando ad inserire una parola di 20 caratteri riscontriamo la dicitura “zsh:segmentation fault” segnalando un errore di Buffer Overflow perché il campo inserito è maggiore di quello impostato nel programma

```
(kali㉿kali)-[~/Desktop]
$ ./BOF
Si prega di inserire il nome utente:prova1_prova2_prova3
Nome utente inserito: prova1_prova2_prova3
zsh: segmentation fault ./BOF
```

- Abbiamo provato a modificare l’array dei caratteri nella variabile char buffer inserendo il valore 30

```
GNU nano 8.0
#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;
}
```

- Inserendo una parola inferiore od uguale ai 30 caratteri (come in foto), non si riscontra nessun errore perché il vettore configurato è impostato per l’inserimento di un massimo di 30 caratteri

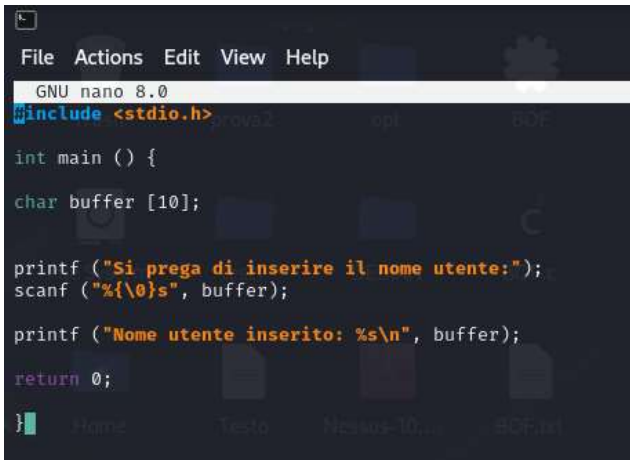
```
(kali㉿kali)-[~/Desktop]
$ ./BOF
Si prega di inserire il nome utente:urtghfjdkeldsiekdjrudyshejdnmcv
Nome utente inserito: urtghfjdkeldsiekdjrudyshejdnmcv
```

- Provando ad inserire una parola maggiore di 30 caratteri riscontriamo la dicitura “zsh:segmentation fault” segnalando un errore di Buffer Overflow perché il campo inserito è maggiore di quello impostato nel programma

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

Remediation Action

Per correggere l'errore dobbiamo inserire un delimitatore di zona `{0}` nell'inserimento della variabile `"%s"`. Abbiamo salvato il file con la nuova modifica.



```
GNU nano 8.0
#include <stdio.h>

int main () {

char buffer [10];

printf ("Si prega di inserire il nome utente:");
scanf ("%{0}s", buffer);

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

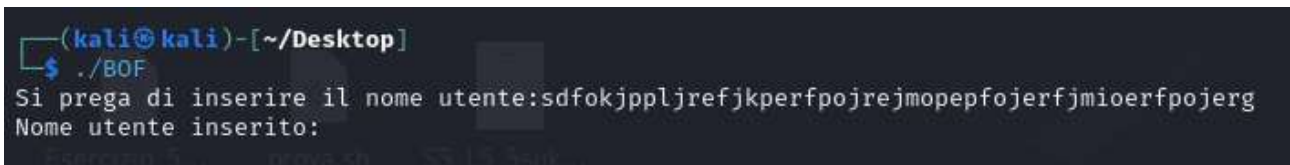
return 0;
}
```

Abbiamo ricompilato il file



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

Provando ad inserire più di 30 caratteri non visualizziamo alcun errore



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