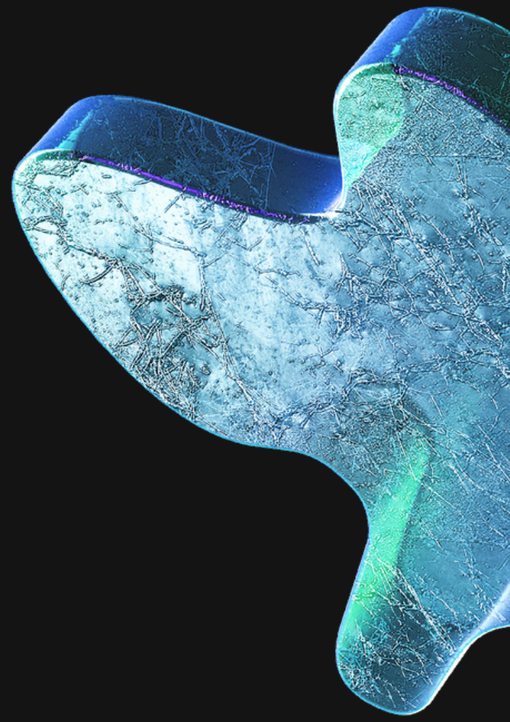
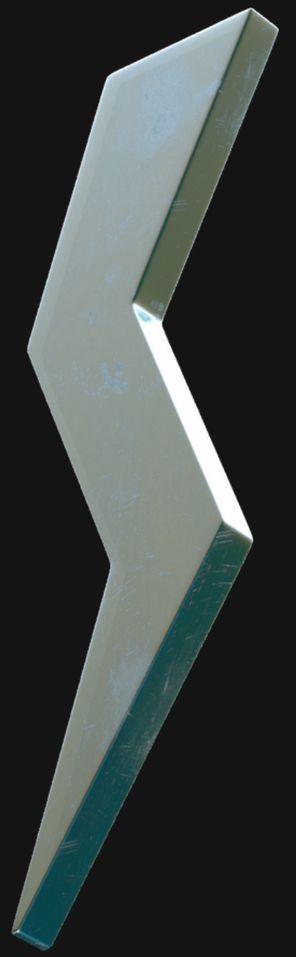
S7/L4



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. 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).

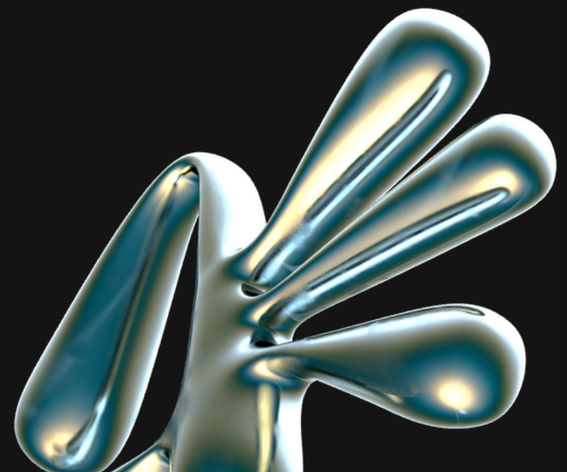


Nell’esercizio di oggi per prima cosa ci viene chiesto di riportare un codice in “C”, un linguaggio di programmazione compilato, il quale ci permetterà di inserire un nome utente che a sua volta verrà ristampato a schermo all’interno di una frase.
Come possiamo notare però questo codice presenta un buffer dedicato all’input dell’utente di soli 10 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;

}

```

Il che vuol dire che nel caso in cui l'input dell'utente sia superiore a 10 si incorrerebbe nel buffer overflow, in questo caso notiamo che il programma ci restituisce "segmentation fault" per un input superiore a 10. Il buffer overflow è una condizione che si presenta quando un programma cerca di scrivere su parti della memoria non riservati a lui.

```
┌──(kali㉿kali)-[~/Desktop]
└─$ ./prova
si prega di inserire il nome utente:1234567890
nome utente inserito: 1234567890

┌──(kali㉿kali)-[~/Desktop]
└─$ ./prova
si prega di inserire il nome utente:123456789012345678901
nome utente inserito: 12345678901234567890
zsh: segmentation fault  ./prova
```



L’esercizio ci richiede di aumentare questo buffer a 30, il che sicuramente ridurrà le possibilità che il buffer overflow si presenti, ma sposterà il problema poco più in là, ci sono vari modi per evitare queste situazioni in base alle situazione in cui ci troviamo.

```
┌──(kali㉿kali)-[~/Desktop]
└─$ ./prova
si prega di inserire il nome utente:12345678901234567890123456789
nome utente inserito: 12345678901234567890123456789

┌──(kali㉿kali)-[~/Desktop]
└─$ ./prova
si prega di inserire il nome utente:123456789012345678901234567890123456789
nome utente inserito: 123456789012345678901234567890123456789
zsh: bus error  ./prova
```

```
5 char buffer [30];
```