Que es un Canary?

El canary es un valor que se coloca en el stack para detectar **buffer overflows**, es decir para evitar que el usuario pueda pisar nuestra pila con un input dado

1. Al entrar a una funcion, el compilador guardda un valor especial (canary) **entre las variables locales y el frame pointer/retorno**
2. Antes de salir de la funcion, se compara el canary original con el que quedo en el stack
   - **Si cambio** → significa que hubo un overflow → **se aborta el programa**
   - **Si sigue igual** → la funcion **retorna normalmente**

Ejemplo funcion vulnerable sin Canary

```
vulnerable:
    push   ebp
    mov    ebp, esp
    sub    esp, 16    ; reservar 16 bytes para buffer local

    ; ... -> (funciones sin limite, en las que puede haber OF)

    mov    esp, ebp
    pop    ebp
    ret
```

*En caso de que alguien meta mas de 16 bytes, se va a pisar el **ebp guardado y el ret**. Entonces podria haber una ejecucion de codigo arbitrario*

Ejemplo funcion vulnerable con Canary

Ahora vamos a proteger la pila, en caso de querer hacer esto en GCC, podemos usar el flag `-fstack-protector`, y nos generaria algo asi:

```
vulnerable:
    push    ebp
    mov     ebp, esp
    sub     esp, 20                          ; 16 bytes de buffer + 4 del Canary
    mov     eax, DWORD PTR __stack_chk_guard  ; Cargar el Canary
    mov     DWORD PTR [ebp - 4], eax          ; Guardar el canary al final del stack

    ; ... -> (funciones sin limite, en las que puede haber OF)

    ; luego de la funcion sin limite, checkeamos el valor del canary para ver si se
piso o podemos retornar correctamente
    mov     eax, DWORD PTR __stack_chk_guard  ; valor original del canary
    cmp     eax, DWORD PTR [ebp - 4]          ; comparamos el original con lo que
tenemos en el stack
    jne     .stack_smash_detected

    mov     esp, ebp
    pop     ebp
    ret

.stack_smash_detected:
    call    __stack_chk_fail                  ; abortamos el programa
```