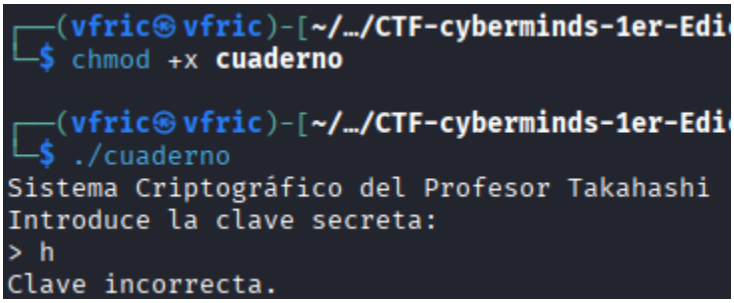
**El Cuaderno del Profesor Takahashi**

Antes de ejecutarlo, se identifica el tipo de archivo con file:

```
└─$ file cuaderno
cuaderno: ELF 64-bit LSB pie executable, x86-64, version 1 (SYSV), dynamically linked, interpreter /lib64/ld-linux-
x86-64.so.2, BuildID[sha1]=4ed93d9219c261e3e9da083086fc5c60e8f9a4b6, for GNU/Linux 3.2.0, not stripped
```

Esto indica que es un binario ELF de 64 bits, no stripped, lo que facilita el análisis estático.

Se otorgan permisos de ejecución y se prueba el programa, El binario compara la entrada del usuario con una clave interna antes de mostrar la flag.




Dado que el binario no está ofuscado, se utiliza strings para buscar texto legible incrustado:

```
└─$ strings cuaderno
/lib64/ld-linux-x86-64.so.2
mgUa
fgets
stdin
putc
puts
strlen
stdout
__libc_start_main
__cxa_finalize
printf
strcmp
libc.so.6
GLIBC_2.2.5
GLIBC_2.34
_ITM_deregisterTMCloneTable
__gmon_start__
_ITM_registerTMCloneTable
PTE1
u+UH
Flag:
profesor2025
Desencriptando...
Clave incorrecta.
Sistema Criptogr
fico del Profesor Takahashi
Introduce la clave secreta:
q~dLge
;*3$"
GCC: (Debian 14.3.0-5) 14.3.0
Scrt1.o
```

La presencia de profesor2025 sugiere fuertemente que se trata de la clave secreta hardcodeada utilizada por el programa. Aunque el programa aparenta realizar un proceso criptográfico (“Desencriptando...”), en realidad la clave está almacenada en texto plano dentro del binario, el programa valida la entrada con una comparación directa (strcmp) y si la clave es correcta, se ejecuta una función que muestra la flag.

No es necesario realizar debugging ni desensamblado avanzado, ya que la información crítica es accesible mediante análisis estático básico.

```
└─$ ./cuaderno
Sistema Criptográfico del Profesor Takahashi
Introduce la clave secreta:
> profesor2025
Desencriptando...
Flag: FIS{PR0F_T4K4H4SH1_CU4D3RN0}
```