Instrucciones para la práctica final del curso:

El objetivo es sencillo: ¡Captura la bandera!

Servidor: *tretornesp.com*

Puerto: *5000*

Permite hasta 5 conexiones simultáneas, por favor, no dejéis terminales conectadas sin utilizar pues bloquearéis a vuestros compañeros.

Preparación de la práctica:

Descarga el código desde:

https://github.com/TretornESP/bec/blob/main/final/reto/server/pwnme/files/main.c

https://github.com/TretornESP/bec/blob/main/final/reto/exploit/exploit.py

Contenidos:

- exploit.py: Esqueleto del exploit.
- main.c: código vulnerable que se ejecuta en el servidor.

Requisitos:

- Máquina Linux, preferiblemente con Ubuntu20.04.
- Gcc y gdb (recomendado ghidra).
- Python3 con pwntools (pip install pwntools)

Compilación:

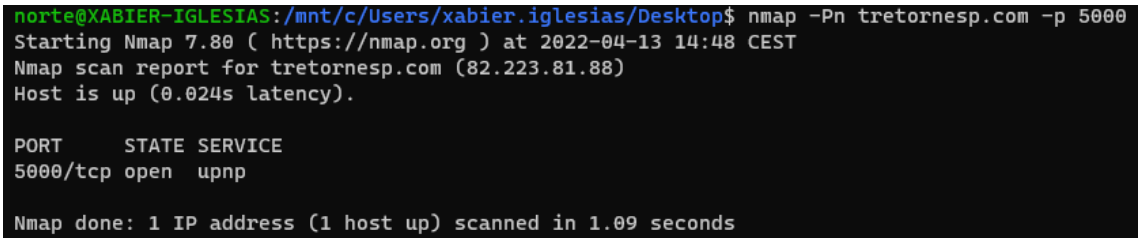
gcc -no-pie -fno-stack-protector -z execstack -g main.c -o a.out

Pasos para completar la práctica:

1) Ejecutar nmap sobre la máquina con el siguiente comando:

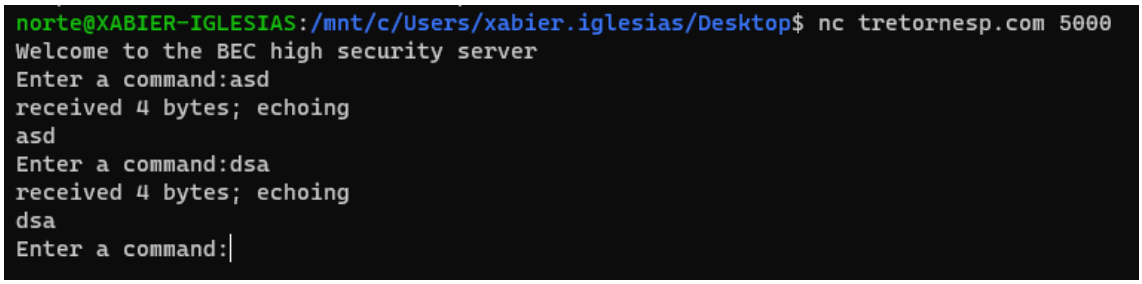
   nmap tretornesp.com -Pn -p 5000

   y comprobar que el puerto esté abierto:



2) Conectarse al servidor mediante netcat y jugar un poco con él:

   nc tretornesp.com 5000



3) Revisar el código fuente del programa (o intentar la ingeniería inversa!).

4) Identificar la función vulnerable y probar que el programa crashea.

5) Seleccionar el payload que deseamos, tenemos principalmente dos opciones:

- Ejecutar una Shell.
- Leer la flag.

Se recomienda usar msfvenom para generarlo.

Nota: No uséis payloads de conexión remota, la máquina solo tiene expuesto el puerto del server.

6) Explotar el programa local dentro de gdb.

7) Modificar el esqueleto del exploit proporcionado.

En concreto necesitaréis 4 cosas:

- Introducir un shellcode válido.
- Calcular el relleno hasta el desbordar el RIP salvado.
- Calcular la dirección de inicio de vuestro shellcode.
- Introducir el comando vulnerable.

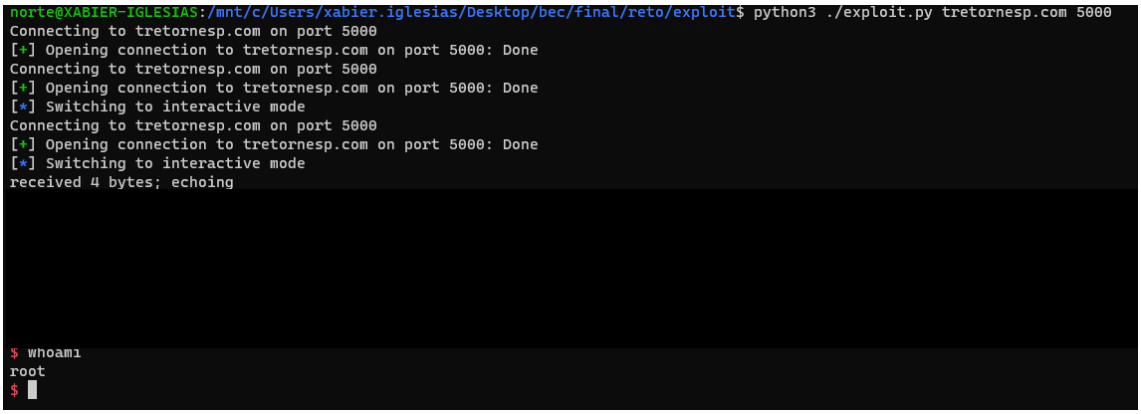
Consejos:

- Vigilad que el shellcode no sea más grande que el buffer!
- Cuidado con los badchars y nullbytes.
- Aslr está desactivado, además tenéis una ayudita para la dirección de inicio, revisad el output de la función vulnerable.
- Si no sois capaces de acertar la dirección del stack, podéis usar una nopsled.

```python
from pwn import *
import sys
import time

if len(sys.argv) < 3:
    print("Usage: exploit.py ip port")
    sys.exit(0)

buf = b'' #Completa el payload!
buf += b"[redacted]"
buf += b"[redacted]"
buf += b"[redacted]"
buf += b"[redacted]"
buf += b"[redacted]"
buf += b'[redacted]'
buf += b'[redacted]'
buf += b'[redacted]'
buf += p64(0x7[redacted]) #Dirección del shellcode

print("Connecting to {} on port {}".format(sys.argv[1], sys.argv[2]))
conn = remote(sys.argv[1],int(sys.argv[2]))
conn.recvline()
conn.recvuntil(b':', drop=True)
conn.sendline(b'[redacted]') # En que comando está la vulnerabilidad? modifica esta linea
time.sleep(1) #Esto es importante! si no los dos send se mezclan
conn.sendline(buf)
time.sleep(1)
conn.interactive() #Si tu exploit hace execve(/bin/bash), si es un shell remoto no hace falta
```

8) Ejecutar el exploit y comprobar que tenemos una Shell:

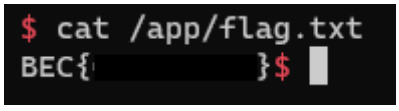
python3 ./exploit.py tretornesp.com 5000



9) Obtener la flag:

La flag está almacenada en el directorio: */app*

cat /app/flag.txt



10) Enviar la flag:

Finalmente, para obtener el certificado, solo tenéis que mandarme la flag y la hora en la que la habéis conseguido a la dirección:

xabier.iglesias.perez@udc.es

Dudas y problemas: Para cualquier duda o problema a la hora de completar la práctica, escribid a la dirección de arriba. Habrá una sesión dedicada a resolver dudas el día 19.

Plazo: Mes de Abril.