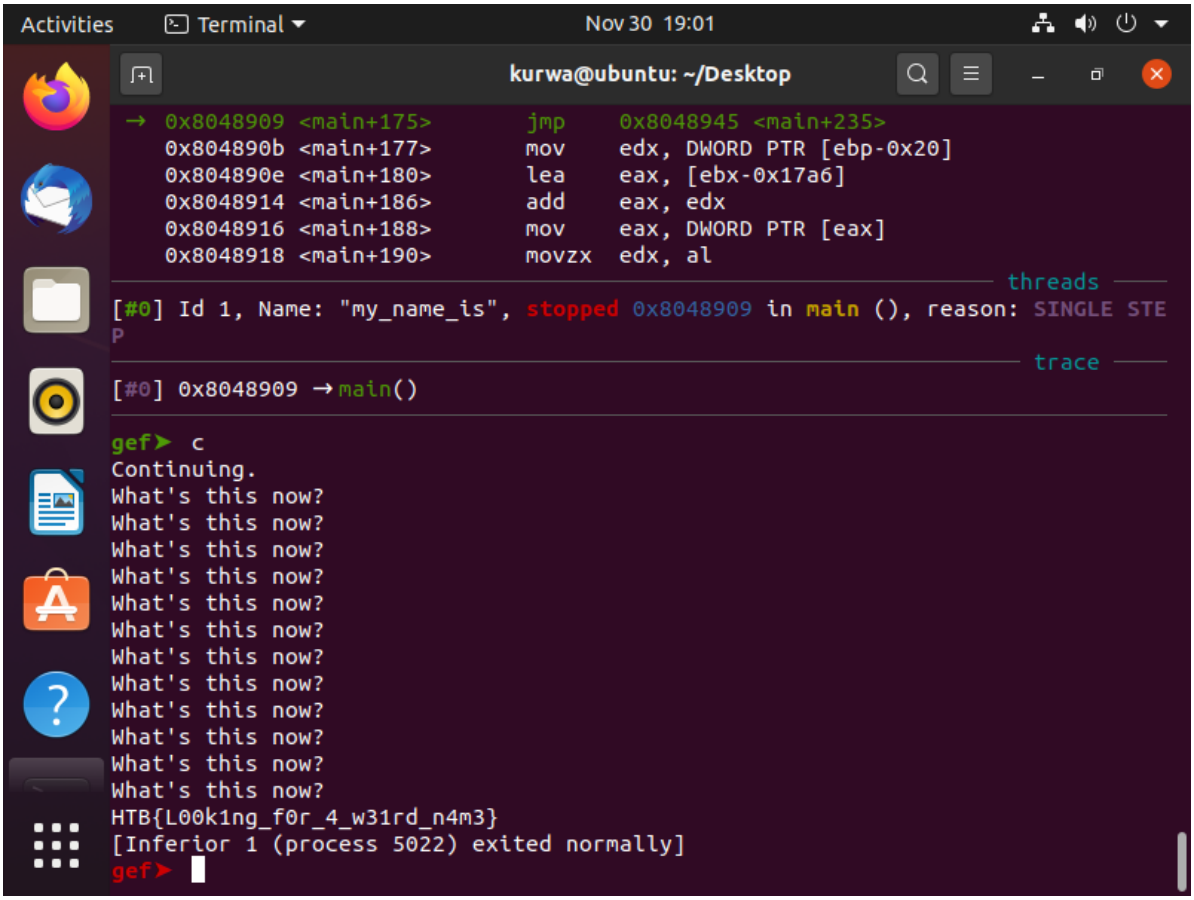
What is it?

A simple reversing challenge which wants you to have a specific username and prevents you from running the binary in specific situations, i.e. during debugging.

```
9
10  puts("Who are you?");
11  uid = geteuid();
12  v5 = getpwuid(uid);
13  if ( ptrace(PTRACE_TRACEME, 0, 0, 0) )
14  {
15    puts("This doesn't seem right");
16    exit(1);
17  }
18  if ( v5 )
19  {
20    s1 = v5->pw_name;
21    if ( !strcmp(v5->pw_name, username) )
22    {
23      for ( i = 0; i < (char *)marker - (char *)main; ++i )
24      {
25        if ( (unsigned __int8)*(_DWORD *)((char *)main + i) == breakpointvalue )
26        {
27          puts("What's this now?");
28          exit(1);
29        }
30      }
31      v7 = strlen(encrypted_flag);
32      s = (char *)malloc(4 * (v7 + 1));
33      decrypt(s1, encrypted_flag, s);
34      s[v7] = 0;
35      puts(s);
36    }
37    else
38    {
39      puts("No you are not the right person");
40    }
41    exit(0);
42  }
43  puts("?");
44  exit(1);
45}
```

How to solve it?

As you can see above, we have a few conditions on which the process will exit. Rather than bothering with those specifically, I just patched the *exit(1)* calls to *NOP*'s so that nothing happens if the conditions for the if statements are met. Subsequently, we see a *strcmp* call that wants us to have a specific username, this is: **~#L-:4;f**. What we can do is to breakpoint when the variable *s1* is set, and edit our username to fit our needs. By using gdb, editing memory, then running the application, we get the decrypted flag as output.



FLAG: HTB{L00k1ng_f0r_4_w31rd_n4m3}