Spot the defect!

```
#include <stdio.h>

int main(int argc, char* argv[])
{   if (argc > 1)
      printf(argv[1]);
    return 0;
}
```

This program is vulnerable to format string attacks, where calling the program with strings containing special characters can result in a buffer overflow attack.

Format string attacks

Type of memory corruption discovered in 2000

- Strings can contain special characters, eg `%s` in

  ```
  printf("Cannot find file %s", filename);
  ```

  Such strings are called format strings

- What happens if we execute the code below?

  ```
  printf("Cannot find file %s");
  ```

- What can happen if we execute

  ```
  printf(string)
  ```

  where `string` is user-supplied ?

  Esp. if it contains special characters, eg %s, %x, %n, %hn?

Format string attacks

If attacker can control malicious input s to printf(s)
then this can

- *read* the stack

  %x reads and prints bytes from stack

  so input %x%x%x%x%x%x%x%x%x%x%x%x%x%x%x%x%x%x
  %x%x%x%x%x%x%x%x%x%x%x%x%x%x%x%x%x%x%x%x%x%
  x%x%x%x%x%x%x%x%x%x%x%x%x%x%x%x%x . . .

  dumps the stack, including passwords, keys,… stored on the stack

- *corrupt* the stack

  %n writes the number of characters printed to the stack

  so input 12345678%n writes the value 8 to the stack

- *read arbitrary memory*

  a carefully crafted input string of the form

  \xEF\xCD\xCD\xAB %x%x...%x%s

  print the string at memory address ABCDCDEF

Preventing format string attacks is EASY

1. Always replace `printf(str)`
   with `printf("%s", str)`

2. Compiler or static analysis (SAST) tool could warn if the number of arguments does not match the format string

As e.g. in `printf ("x is %i and y is %i", x);`

`gcc` has (too many!) command line options to get such warnings

```
-Wformat -Wformat-no-literal -Wformat-security...
```

But: if the format string is not a compile-time constant, we cannot decide this at compile time ☹

Would you want your compiler or SAST tool to give a false positive or a false negative in such cases?

Check https://cve.mitre.org/cgi-bin/cvekey.cgi?keyword=format+string
to see how common format strings still are