Low level programming bugs

Format String Bugs
- Allow arbitrary memory writes

```c
printf(char* fmt, …);

Can do printf(str);
As long as str doesn't contain any %
to print char* str out.
```

Bug in Wuftp 2.6.0

```c
void loguser(char* user)
{
    char buf[512];
    snprintf(buf, sizeof(buf), user);
    ...
}
```

- ARGP points to the expected first argument (right above user
  on the stack)
- If user enters a string with % in it they can go up the stack even
  if there are no extra varargs supplied
- Suppose we have `printf("%s%n%d", str, &cnt, x);`
  - %n writes the number of printed bytes so far to memory
    location specified in argument
  - Make ARGP point to a desired location in buf, it reads the
    memory address to write the %n value to. This allows
    arbitrary writes to any location.
- Can control count with things like %100000d
- user=\"0000\xf1\x23%61255d%n<shellcode>\"  
  - Count gets set to 61261
  - ARGP is pointing to 0xf123 in buf, interprets it as the
    location to write %n value to, so writes 61261 to 0xf123
- After snprintf hits the requested number of bytes to print it
  stops writing output but will keep running.