# What's So Funny?
# CIS 371 Security in Computer Science

Brian C. Ladd

Fall 2022

## Learning Outcomes

After completing this assignment, a student should be able to

- Examine a Linux executable.
- Use `man` and on-line documentation to find new (to them) tools.
- Exploit particularly bad programming practice to get to "hidden" contents.

## Procedure

1. Read the assignment.

2. Get `whatsSoFunny`.

   The `whatsSoFunny` is a file in the assignment directory for this assignment. You should download a copy.

3. Bet you wonder what *kind* of file you just downloaded. Just guessing, but there is probably a command for that. Your favorite search engine can get you started on that one.

   > **Include**
   >
   > Give the commandline you used to get the type of `whatsSoFunny` and the answer that you got.

4. So, if you want to run an executable program on Linux, it must have the executable permission bit set. If yours is a well-behaved browser, that bit is not set for the file you just downloaded.

   Note: making a random file executable is not good security practice. `whatsSoFunny` is not *malicious* (it was not written with bad intent) but you are trusting that it does not call the `unlink` system call to remove all the files you have write access to on your computer when you run it.

> **Include**
>
> Show the initial permissions `whatsSoFunny` had when it was downloaded, the command you ran to change its permissions, and the final permissions it has.

5. Trusting `whatsSoFunny`, run it.

6. If `whatsSoFunny` is checking what you typed against some password(s) *and* it does not use the network (it does not) where could that password be stored?

> **Include**
>
> Think about this and include an answer.

7. Assuming you did not guess the password right off (if you did, go to the next step), the program gave you a hint.

   You could look at the whole program using a tool like `less`. But much of the program is not *printable*. `less` will show hex codes for bytes that do not have a printable representation.

   It would be nice not to go through 13 or so pages of bytes, hoping to spot a useful string. Finding a more focused tool would be nice.

   Rather than an on-line search engine, try the features of `man`:

   - `-k` / `--apropos` searches for **keywords** you provide in the short description of the manual pages.
   - The apropos search can be limited to certain sections of the manual with the `-s` (comes after the `-k`) switch which takes a number. Section 1 is where executables (tools, Linux programs) are documented (rather than, say, `C`/`C++` functions or system calls).

> **Include**
>
> What commandline did you run to find a tool as suggested in the hint? What is the name of the tool.

8. Assuming you found a candidate password, run `whatsSoFunny` and try it. `whatsSoFunny` does `not` keep track of how many guesses have been made.

> **Include**
>
> Include the commandline you ran to look inside `whatsSoFunny`.
> What's so funny? Include the joke in your answer file.

## Submit through Classroom Management System

Submit your answer file through BrightSpace.

Note: there is only one version of this assignment across the class. Please do not give away the joke (on the steps to get there). You may help others in general.