Week 1 - Supplementary Materials

[Markdown](#) | [PDF](#) | [MS Word DOCX](#) | [Libre ODT](#) | [HTML](#)

The Command Line Interface (CLI) and Filesystems

A command line interface accepts text commands and executes programs.

Using a CLI was once the only way a person could operate a computer. Today, most users interact with a graphical user interface (GUI) that allows you, for example, to click an icon on your desktop with a mouse or tap on a smartphone touchscreen.

This week we focus on using a CLI to interact with and navigate a filesystem. This is a basic skill required for hackers.

Try a CLI in your browser:

- [BROWSER CLI](#)

Videos:

- Filesystems [MP4](#) | [WEBM](#)

- Navigating Directories [MP4](#) | [WEBM](#)

- Manipulating Files [MP4](#) | [WEBM](#)

Basic Commands

`pwd` - print working directory

Use the pwd command to see what directory you are currently in.

On most systems, users start a session in their home directory, which is the directory set by the system administrator for that user. Your home directory is usually `/home/user` where "user" is your account's username. For the administrative superuser named "root", a concept we will explore later, the home directory is `/root`

cd - change directory

To navigate to a specific directory use the `cd` command. The `cd` command takes a path as an argument such as `cd /usr/bin` and moves you into that directory.

If you would like to navigate up the directory tree, use `cd ..` If you are in `/usr/bin` then the `cd ..` command will bring you to `/usr/bin` If you try `cd ..` again, it will bring you to `/` which is the absolute top of the directory tree. This is often called the "directory root", not to be confused with the administrative superuser named "root".

To navigate back to your home directory, use the `cd` command with no arguments or `cd ~` where ~ means your home directory.

`ls` - list

`ls` without any arguments will list all the files and directories within the directory you are currently in.

Pass a specific directory as an argument to `ls` to see the contents of that directory. For example, `ls /usr/bin` will list the contents of /usr/bin

If you use the `-l` option or "switch" with `ls`, you will see much more detail about the directory contents. Try: `ls -l /usr/bin`

To make directory listings easier to read, you can "pipe" the output to `less` such as `ls -l /usr/bin | less` We will talk about pipes and input/output in more detail in later classes.

If you try `ls --help` you can see all options available for `ls` and most other commands.

clear - clear screen

You may find that your CLI has become cluttered. To clear your screen use `clear`.

To clear your total terminal / CLI history, try `history -c`

man - manual

The `man` command will open the usage manual, if one exists, for the command that you pass to it as an argument. For example, `man ls` will open the manual for the `ls` command. You can quit this manual view by typing q

Hackers use the slang [RTFM](#) to refer to the tendency for users to ignore manuals.

Often, you can view a shorter version of the options for a command that is similar to the manual with the `--help` or `-h` switches, such as `ls --help` or `nano -h`

mkdir - make directory

To create a new directory, use the `mkdir` command. You can only create a directory where you have permission to do so. Permissions are a concept we will cover in class later. For now, try creating the directory `catphotos` in your home directory: `mkdir ~/cats`

touch - create an empty file

The touch command creates a new, empty file such as `touch ~/cats/awesome-cat-names.txt`

nano - simple text editor

Nano is one of many text editors that can be loaded with the CLI, but is probably the simplest. It will load a file in a separate view that can be exited with CTRL+X and you will see this at the bottom of the screen represented as ^X For more information, [read this tutorial](#).

Create your own list of awesome cat names with `nano ~/cats/awesome-cat-names.txt`

Traditionally, the two text editors that hackers use are `vi` or `emacs` There is a tongue-in-cheek ["editor war"](#) between these two editors.

mv - move a file

mv can move a file or directory somewhere else on the filesystem but **is also used to rename files**. This is because renaming a file is just creating a new entry for the file, the same operation as moving it to another directory, as far as the filesystem is concerned.

Use the `mv` command and pass two arguments, the first being the current filename and the second being the new name: `mv awesome-cat-names.txt cool-cat-names.txt`

Use `mv -R` to move a directory and its contents.

Be careful! If there is already a file with the same name, you will overwrite that file.

cp - copy

To copy a file use the cp command: `cp cool-cat-names.txt cool-cat-names2.txt`

Use `cp -R` to copy a directory and its contents.

rm - remove

Delete a file: `rm cool-cat-names2.txt`

Use `rm -R` to delete a directory and its contents.

Be careful! There are jokes on the Internet that hackers like to test on new users, or "n00bs", such as `rm -Rf /` that can be very dangerous. Luckily, you usually have to be the administrative superuser "root" to do serious damage to your filesystem.