Lab 09: When disaster strikes

Hands-on Unix system administration DeCal

Due October 29, 2012, at 6:10pm

Nifty networking tools

1. `ifconfig` is a handy utility used to review and configure network interfaces. Using your ocf account, run `ifconfig` on tsunami.
   (a) List all the currently active interfaces on tsunami.
   (b) What is tsunami's MAC address? IP address?
   (c) What do you think the loopback interface is used for? (`lo`)
2. You've already seen `netcat` used several times this semester. Describe how you can transfer a file between two hosts using `netcat`. Be sure you describe the commands used both on the client and the server. Hint: see `man netcat`, in particular -l, -p options. And use pipes!) And why is this a bad idea to do this if you are transferring particularly important data?

Even moar useful utilities

1. Using your ocf account, look up tsunami's current load averages. What are they? How long has tsunami been up? (Obviously, answers will vary)
2. Describe the command you would use to create a compressed archive of your home directory, using `tar`.
3. Describe how you would use `scp` to transfer this archive to your home directory on tsunami.
4. Review time! Devise a pipeline that extracts the username, idle time, and what program that user is running using `w`. (Hint: you can use tr and cut)

Patching with `diff` and `patch`

The commands `diff` and `patch` are often used to obtain the *differences* between original files vs. updated files in such a way that people who only have the original files can turn them into the updated files with just a single patch file (that contains only the differences). If that sounded horribly confusing, don't worry, read on!

The most simple way of using `diff` is getting the differences between two files (an original file and an updated file). For example, you could write some words in a simple text file, make some modifications, and then save the modified content to a second file. Then, you could compare these files with diff, like this (try it!):

```
$ diff original_file updated_file
```

You can use diff with the `-u` flag to tell `diff` to output the differences in *the unified format.* This output format is often used as input to the `patch` program.

If we want to create a patch, we should redirect the output of `diff` into a file.

```
$ diff -u original_file updated_file > patchfile
```

1. Describe how you would obtain the differences between two files, named `original.c` and `updated.c`, and create a patch file out of those differences.

2. Now that you know how to create a patch with `diff`, describe how you would apply that patch to `original.c` (Yes it sounds silly. Of course, in real life it wouldn't make that much sense to apply the patch on the files we created the patch from.) with the command `patch`. (Read up on `patch`'s man page)