Linux Lab 10 - Finding Files and Content

Reading

The Linux Command Line, chapter 17, "Searching for Files", pp. 185 - 200 (printed version) or pp. 217 - 233 (pdf).

Lab

which

We've already covered one handy program for finding executables called `which`. Remember that `which` will search the directories in the $PATH variable and tell us which path and file will be executed.

locate

Linux keeps a database of files and pathnames which is searchable and fast. You can search the database using the `locate` program. The major limitation of locate is that the database is updated only once a day or so. If the database is old, it may not find the files you want.

1. Enter the commands on page 188 (print version) or 217-218 (pdf) of the text. Make sure that you understand why the two commands are different. Also, notice the speed of the results so you can compare it to the `find` command.
2. Create a new file in your home directory. There are lots of ways to do this, but the `touch` command is handy. It updates the file access times and creates a new file if one doesn't exist. `touch umptyfratz` (you can name the file whatever you like.)
3. Use the `locate` program and try to find the file you just created. It should fail.
4. Update the `locate` database with the command:
   `sudo updatedb`
5. Try to find the file using `locate` again. It should succeed this time.

find

The `find` program is powerful, but it can also be complicated. It can search on just about every attribute a file can have. Be careful, however. Using `find` on a production server with a search starting from the root can take enough resources to impact the server operation. It will be fine on our little VMs, though.

1. Use the find command to see if there are any .mp3 or .mov files on your Ubuntu image

The find command can generate a lot of errors, especially if you are searching from the file system root "/" and do not have root privileges. You can redirect the errors to nowhere by adding `2>/dev/null` to the end of your command. You can also reduce errors by using `sudo`.

The `find` program can search for usernames and permissions. This is very handy for finding files that have the SetUserID (suid) or SetGroupID (sgid) permissions set and are owned by root. The octal mode with only suid set is 4000, and with only sgid set it is 2000. The -perm test can find these, but it has a huge gotcha. If you use -perm 4000 it will find only files with exactly 4000, but hardly any files are set that way. For example, the permissions on the `passwd` program are 4755, or -rwsr-xr-x, and -perm

4000 won't find it. However, you can change this behavior by using `-perm -4000` (note the dash before 4000). In this case it will find any file that has the suid bit set regardless of the other settings. Likewise, `-perm -2000` would find files with the sgid bit set. To find files that have the suid bit set and are owned by root (these are the dangerous ones) you would use:

```
find / -user root -perm -4000
```

2. Use the command above to find the commands that have suid root.
3. It would be nice to have a file that had listed all the suid root files that are normally present, and list them in `ls -l` format. The `-exec` option of find will allow you to do that. There is an example of the `-exec` option in use on page 197 (print) or 228 (pdf) of the text. Note that the `'{}'` part tells `exec` where to insert the results from find into the command you are executing. Also note that the command ends with `';'` which tells `find` that the command is finished. Use the command from step one and add the `exec` option to the end as in the first example on page 197/228. Once that is working, redirect the results to a file. This may be nice to have for future CyberPatriots images.

The exec option is very powerful, as it can execute almost any command. It doesn't just have to execute ls -l on the files it finds.

grep

The programs `which` and `locate` look for file names. The program `find` looks for file names, and just about any file attribute (size, date, permissions, etc.) you can think of. If you are looking for the **content** of a file, you need the powerful tool called `grep`. The format of the command is:

```
grep [OPTION]... PATTERN [FILE]...
```

We've used grep already in ways like `netstat -na | grep tcp`. This takes the output of the netstat command and pipes it into grep. Then grep filters out everything except the lines containing the PATTERN tcp.

If we want to find file content, we use the form `grep PATTERN FilePath`. There is a helpful option we can use when searching for files, -r, where r means recursive. It will search the directory we list in FilePath, and all subdirectories of FilePath. (Note: It is not usually a good idea to search recursively from the file system root, i.e. `grep -r somestuff /` . The grep command will have to examine the content of every file on the computer, which takes a lot of resources and time.)

4. Look to see if the word "scrub" is in any file in the /etc directory or any of the /etc subdirectories. Note: there is no significance to the word scrub. It is just a word that will give you a few answers, but not fill your screen to overflowing.

Turn in

Take the file you generated in step 2 of the find command, and pipe it into `wc -l`. This will count the lines in your file and tell us how many files are suid root. How many files do you have that are suid root?