Linux Lab 3 Files and Permissions

Read

Read “The Linux Command Line”, Chapter 9, pp 77 - 93 in the printed text or pp 90 - 109 in the pdf. Feel free to run any of the examples in a terminal on your Ubuntu VM.

Read or listen to CyberAces Module 1 Linux, Session 6, Files and Permissions
(https://tutorials.cyberaces.org/tutorials/view/1-1-6.html)
Do the exercises in the CyberAces module!

Hand In

Hand in your commands and answers to questions 4, 8, and 11

Exercises

1) If there isn't an extra user on your computer, create one now. I'll call mine user1.

2) As root, create a directory in /var called test.

3) As root, create a group called test.

4) Use ls -l on /var and look at the file rights that are on /var/test. Pay special attention to:

   What is the user (owner) and what permissions does it have?

   What is the file's group, and what permissions does the group have?

   What permissions does the world (other or everyone) have?

5) Use commands to change the file rights on /var/test so that members of the group test have read/write/x access. (hint: you will need to create the test group, then use chgrp (or chown with the : option) to make test the group that owns /var/test instead of root, then chmod if the permissions need to change.)

6) Make user1 a member of the group test. (hint: either gpasswd, or usermod)

7) Use the su command to switch user to user1. Create files in the /var/test directory. Check the file rights with ls -l.

8) Exit from su to get back to your regular user. Can your regular user delete files in /var/test? Can it read them? Why? (hint: ls -l, and the command whoami may help. Is your user the user/owner, group, or other?)

9) Change the rights for the /var/test directory, and also the files within it, so that only the user/owner and the group have read and write access. Everyone else/other should have no access at all. (hint: chmod ### is fastest. To change all files within test, you'll need to use the recursive flag for chmod.)

10) Test your changes by trying to read files in /var/test with your regular user.

setuid and setgid

Read

https://linuxconfig.org/how-to-use-special-permissions-the-setuid-setgid-and-sticky-bits
https://docs.oracle.com/cd/E19683-01/816-4883/6mb2joatb/index.html

In addition to the rwxrwxrwx bits, there are three more at the beginning that are not as easy to see. They are called setuid, setgid, and sticky. The first two are critical, as they can be used by an attacker to hide scripts or applications that have root privileges and run them as a normal user. The first article explains the basics of setuid and setgid.

If you use octal mode, the permission r-xr-xr-x would be 555. There is another set of 3 bits ahead of these, setuid, setgid, and sticky. So, the permissions for that file, without any of the 3 bits set would be 0555. If the setuid bit were set, it would be 4555.

In CyberPatriot, you may need to search for evil files hiding under setuid. The second article shows you one way to do this and slide 16 of the CyberAces module shows another. You will get a list of files with setuid set. The way to tell which ones should be there and which ones should not is to run the same command on a clean VM you've created and compare the results to the VM you are concerned about.

11) What files on your VM have the setuid bit set? (Save this so you can compare it with other VMs later.)

Hand In

Hand in your commands and answers to questions 4, 8, and 11