Assignment 2
CIS425 – Fall 2017
Point Value: 100 points
Assignment Due Date: **In class Thursday September 21, 2017**

Submission Instruction

Please submit a hard copy of your assignment to the instructor in class on the due day.

Short answers

1. **What is a sector** (3 points)**? What is a cluster (**3 points**)? What is a Volume?** (4 points)
   Sector- is the smallest container a computer can use to store information
   Cluster- multiple continuous sectors
   Volume- logical partitioning of a disk, which consists of one or more clusters

2. **Assume that a user wants to store a file named "test.txt" on a computer. The file's size is 2,500 bytes. A sector can contain 512 bytes. Each cluster is made of 4 continuous sectors. How many sectors are needed to store the file (**5 points**)? How many clusters are needed (**5 points**)?** *Please show your work and you may get partial credits***.**
   1 sector=512 bytes
   2500 bytes=? Sectors
   2500/512 = 4.88 sectors, so 5 sectors are needed to store the file

   4 continuous sectors* 512 bytes = 2048 bytes
   2500/2048 = 1.22, so 2 clusters are needed to store the file

3. **Please calculate the capacity of the magnetic disk given the following specifics (**10 points**):**
   a. 512 bytes/sector
   b. 8 sectors/track
   c. 90 tracks/side
   d. 3 platters

   3*2*90*8*512=2,211,840 bytes

   = 2211.84KB

   =2.2MB

4. **Please explain what is slack space? Why is slack space important to digital forensics examiners?** (10 points)
   Slack space is the difference between the space that is assigned and the space that is actually used. It is important to digital forensics examiners because slack space cannot be accessed by normal user or the operating system.

5. **Please explain what is the metadata of a file (**4 points**)? What are the MAC times that are associated with a file in Linux OS?** (6 points)

   Metadata of a file consists of: file size, allocated blocks, ownership/permissions, and time stamps.
   - Modified time stamp

- Accessed time stamp
- Changed time stamp

6. **Please look at the file's metadata and tell about the following information about this file?**
-rw-rw-r-- 1 tian tian 0 Sep 2 16:45 test.txt

Name of the file (1 point): test.txt
Owner of the file (1 points): tian
Group of the file (2 points): tian
The file permissions for the user (2 points): read(r) and write(w)
The file permission for the group (2 points): read(r) and write(w)
The file permission for the other (2 points): read(r)

7. **Please explain what are the passwd file and the shadow file (**6 points**)? Where is each of them stored (**2 points**)? Why are they very important? (**2 points**)**

Password file- contains each user account information but no password
- Stored:/etc/password

Shadow File- contains password hashes for user accounts
- Stored:/etc/shadow

The shadow files are important because people who try to break into the system won't be able to with a shadow file; since the password will be encrypted/hashed. Password files are also important because they store useful information like: username, User ID, group ID, home directory, and default shell info.

8. **Given the following records, please explain what each field means.**

(1) A record from passwd file (5 points):
testaccount:x:1001:1001:This is a test account:/home/testaccount:/bin/bash

- testaccount = username
- x = placeholder or "dummy value" of the encrypted password located in shadow file
- 1st 1001 = User ID
- 2nd 1001 = Primary Group ID
- This is a test account = comment field
- /home/testaccount = the path of the user's home directory
- /bib/bash = the program to run upon initial login (user's default shell)

(2) A record from shadow file (5 points):

testaccount:6rRkFTUEC$VJpiBPqh8bABn8I89245f:106691:0:999999:7:::

testaccount = username

6rRkFTUEC$VJpiBPqh8bABn8I89245f:106691 = encrypted/hashed password

106691 = number of days since unix epoch(Jan 1970) that the password was last changed

0 = minimum days between passwords

999999 = maximum time pass word is valid

7 = number of days prior to expiration to warn users

::: = reserved for future use

9. **What are the two primary categories of logs (**4 points**)? Please explain what is the "*.bash_history*" file under a user's home directory?** (6 points)

The two primary categories of logs are: User Activity Logs and Syslog. Bash stands for Bourne Again Shell. Commands typed in any shell session are stored in a file in the user's home directory called ".bash_history".

10. **Please explain what will happen in the *directory entry table*, *inode table*, and *the hard disk* after a user enters each command. You need show each step in order to get points. You can draw a diagram to make your explanation easier to understand. (**10 points**)**
    1. **echo "I am file 1." > file1.txt(creates a file titled "file1.txt" in the current directory)**
       directory entry table- The directory entry table stores the name of the file, and the time the file was created.
       inode table- stores the size of the file, when the file was last accessed, and who owns the file.
       hard disk- stores the information contained within the file "I am file 1"

    2. **ln file1.txt file2(creates a hard link to file1 and makes a new file named file2 in the current directory)**
       directory table- stores the name of the file, "file1.txt", file's extension, ".txt" and the time of the file's creation.
       inode table-With this command, file1.txt's link increases 1.
       hard disk- stores the contents of "I am file 1"

    3. **ln file1.txt file3(creates a hard like to file1 and makes a new file named file3)**
       directory entry table- different creation time and attributes of the file
       inode table-stores the file size, when it was last accessed, and who owns the file. The file1.txt like count increases 2.
       hard disk- stores the contents of "I am file 1"

    4. **echo "I am file 4." > file4.txt(creates a file titled "file4.txt" with the text contents " I am file 4") in the current directory.**
       directory entry table-stores the name of the file(file4.txt), the file's extension(.txt), and the time the file was created.
       inode table- stores the file size, when it was last accessed, and who owns the file.
       hard disk-stores the infor contained in "I am file 4". Current number of links in the metadata is 0.

    5. **ln –s file4.txt file5.txt(creates a softlink between file4.txt and file5)**
       directory table- stores the name of the file(file5), the file's extension(.txt), date/time the file was created, and the file's link to plain text, link target: file4.txt.
       inode table-stores the file's size, when it was last acccesed, and who owns the file.
       hard disk-stores the content of the file(" I am file4."

6. **rm file1.txt(removes the file file1.txt's link)**
   The inode table still contains the metadata of file1.txt. The data about file1.txt still exists in the previous locations, however the data is inaccessible because this command remoded the link to the data. Also, file2 says "I am file 1" because it was a copy of another file's data (hard link).

7. **rm file2**
   directory table-(same as 6)
   inode table-(same as 6)
   hard disk-(same as 6)

8. **rm file3**
   directory table-(same as 6)
   inode table-(same as 6)
   hard disk-(same as 6)

9. **rm file4.txt(deletes link to file4.txt, but keeps the data about the file, but makes file5 useless. File5loses its link to file4.txt since it is a soft link.**
   directory table- changes directory because the link to file4.txt is lost. The content of file5 also changes.
   inode table- same, but has no hard link(file4.txt) to point to.
   hard disk- the link is deleted(which frees up a little space). File5 is still registered as being linked to file4.txt.

10. **rm file5.txt**
    directory table-(same as 6)
    inode table-(same as 6)
    hard disk- (same as 6)