Assignment 3 Notes

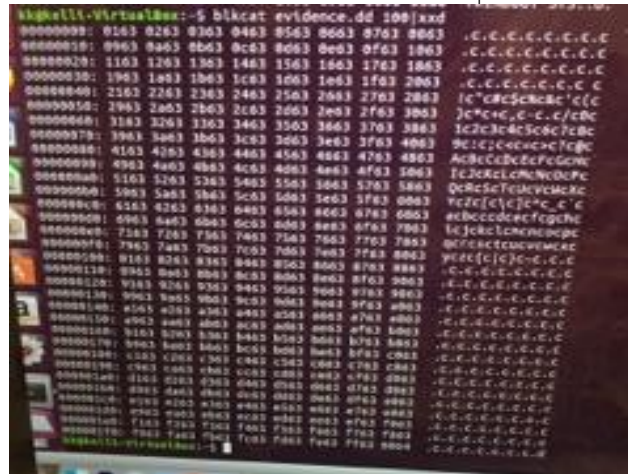Assignment Number: ___3____ Examiner Name: Kelli Kinnikin

Date & Time	Activity
11/1/17: 2:00	First I downloaded the two files from schoology. • scp cis425@mathcswc.fontbonne.edu:~/evidence.dd ./ • scp cis425@mathcswc.fontbonne.edu:~/evidence.md5 ./
11/1/17: 2:05	Then I had to make a directory named 'Evidence' with this command • mkdir Evidence
11/1/17: 2:06	Then I needed to mount it with the command • sudo mount –t auto –o loop,ro evidence.dd Evidence
11/1/17: 2:07	Then I needed to change the directory to Evidence with the command • cd Evidence
11/1/17: 2:07	Then I used 'ls –l' to list all the files in the root directory of evidence.dd
11/1/17: 2:10	Next, I used the 'fstat evidence.dd' command to find information about the disk. • File System Type: FAT16 • Volume: Label(Root Directory) • Label: File System Type label: FAT16 • Sector Size: 512 • Cluster Size: 8192
11/1/17: 2:15	I then used a list of the following commands to determine whether or not the sectors on the disk were allocated or unallocated. • Command: 'blkstat evidence.dd 500' (allocated) • Command: 'blkstat evidence.dd 1000' (allocated) • Command: 'blkstat evidence.dd 5000' (allocated) • Command: 'blkstat evidence.dd 8000' (allocated)
11/1/17: 2:20	After that, I needed to write down the content of the block 0 and 100 on the disk.

Initials_____KK_____
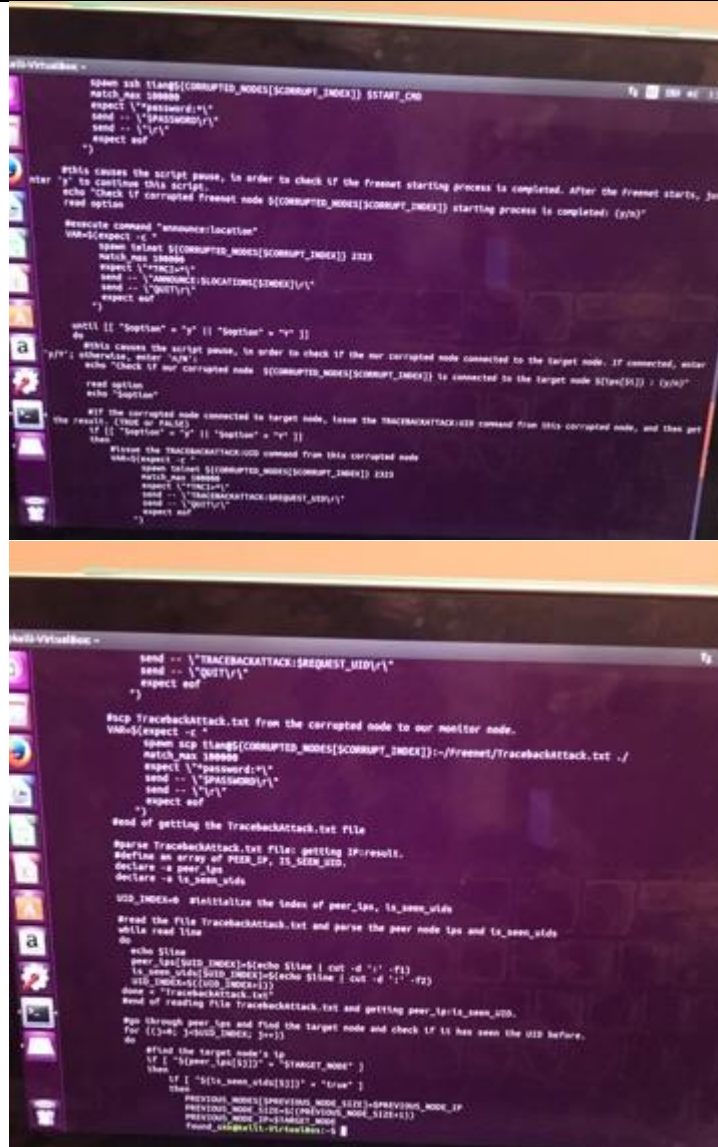
- Command: 'blkcat evidence.dd 0|xxd'

- Command: 'blkcat evidence.dd 100|xxd'

| 11/1/17: 2:30 | Next, I had to find the file named "autotracebackattack.sh" and write down the following information: |

Next, I had to find the file named "autotracebackattack.sh" and write down the following information:

- Command: 'istat evidence.dd 61
    a. inode # = 61
    b. permission = read and write
    c. size (in bytes) = 11,324
    d. last access time = 6/14/2012… 13:39:54
    e. last modified time = 4/27/14…00:00:00
    f. last changed time = 6/14/2012… 13:39:54
    g. whether it is allocated or unallocated : unallocated
    h. data blocks used by this file: 10,879 through 10,894
   i. Content of this file.
- I used the command: 'icat evidence.dd 61'

Initials_____KK_____

```
hi@hill-VirtualBox:~$ icat evidence.dd 61
###############################################################
whatsnew.egit-noteworthy.link-icon = images/logicLabel/git48.png
whatsnew.egit-noteworthy.hover-icon = images/topIcLabel/git48.png
 tedPeers from previousNodeIp to our monitor node.
VAR=$(expect -c "
    spawn scp tian@$PREVIOUS_NODE_IP:~/Freenet/ConnectedPeers.txt ./
    match_max 100000
    expect \"*password:*\"
    send -- \"$PASSWORD\r\"
    expect eof
")

#end of getting previous node's connected peers.

echo "Get connected peers from previous node $PREVIOUS_NODE_IP"

#parse connected peers' IP, PORT, and LOCATION.
#define an array of IPs, PORTS, and LOCATIONS.
declare -a ips
declare -a ports
declare -a locations
declare -a is_forwarded_request  #true means the peer forwarded the file request; false means the peer didn't see the file request
at all.

INDEX=0   #initialize the index of ips, ports, and locations

#read the file ConnectedPeers.txt and parse the ip, port, and location
while read line
do
    echo $line
    ips[$INDEX]=$(echo $line | cut -d ':' -f1)
    ports[$INDEX]=$(echo $line | cut -d ':' -f2)
    locations[$INDEX]=$(echo $line | cut -d ':' -f3)
    INDEX=$((INDEX+1))
```

```
INDEX=0   #initialize the index of ips, ports, and locations

#read the file ConnectedPeers.txt and parse the ip, port, and location
while read line
do
    echo $line
    ips[$INDEX]=$(echo $line | cut -d ':' -f1)
    ports[$INDEX]=$(echo $line | cut -d ':' -f2)
    locations[$INDEX]=$(echo $line | cut -d ':' -f3)
    INDEX=$((INDEX+1))
done < "ConnectedPeers.txt"

echo "Completion of Parsing IP:PORT:LOCATIONS"
#end of getting ip, port, and locations of each connected Peer of the previous node.

#loop onto each of the ConnectedPeers[index], stop Freenet and change its maxOpennetPeers++, then start Freenet again.
for (( i=0; i<$INDEX; i++))
do
    TARGET_NODE=${ips[$i]}

    #check if this connected peer is a previous node.
    IS_PREVIOUS_NODE="false"
    for ((p=0; p<$PREVIOUS_NODES_SIZE; p++))
    do
        if [ "${PREVIOUS_NODES[$p]}" = "$TARGET_NODE" ]
        then
            IS_PREVIOUS_NODE="true"
            break
        fi
    done
    #end of checking if this connected peer is a previous visited node

    #if it is a previous node, skip it.
    if [ "$IS_PREVIOUS_NODE" = "true" ]
    then
        continue
    fi

    echo "Testing target node $TARGET_NODE..."

    #stop Freenet
    STOP_CMD="/home/tian/Freenet/run.sh stop"
```

```
STOP_CMD="/home/tian/Freenet/run.sh stop"

VAR=$(expect -c "
    spawn ssh tian@$TARGET_NODE $STOP_CMD
    match_max 100000
    expect \"*password:*\"
    send -- \"$PASSWORD\r\"
    send -- \"\r\"
    expect eof
")

#change maxOpennetPeers++ in freenet.ini
maxOpennetPeersLine=$(echo "5e" | sed -n '/maxOpennetPeers/p' freenet.ini)
MAX_OPENNET_PEERS=$(echo $maxOpennetPeersLine | cut -d '=' -f2)
NEW_MAX_OPENNET_PEERS=$((MAX_OPENNET_PEERS+1))

new_maxOpennetPeersLine="node.opennet.maxOpennetPeers=$NEW_MAX_OPENNET_PEERS"
sed -i -e "s/$maxOpennetPeersLine/$new_maxOpennetPeersLine/g" freenet.ini

echo "TARGET NODE IS: $TARGET_NODE; PREVIOUS_NODE IS: $PREVIOUS_NODE_IP"

#start Freenet: TARGET NODE
START_CMD="/home/tian/Freenet/run.sh start"
VAR=$(expect -c "
    spawn ssh tian@$TARGET $START_CMD
    match_max 100000
    expect \"*password:*\"
    send -- \"$PASSWORD\r\"
    send -- \"\r\"
    expect eof
")

#this causes the script pause, in order to check if the freenet starting process is completed. After the freenet starts, just e
nter 'y' to continue this script.
echo "Check if freenet $TARGET_NODE starting process is completed: (y/n)"
read option
#end of start Freenet: TARGET NODE

#start one of our corrupted nodes. After it starts, executes command "announce:location"
CORRUPT_INDEX=0 #decide which corrupted node we are going to use for this time.

#start Freenet
START_CMD="/home/tian/Freenet/./run.sh start"
VAR=$(expect -c "
    spawn ssh tian@${CORRUPTED_NODES[$CORRUPT_INDEX]} $START_CMD
```

Initials_____KK_____

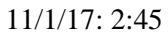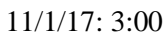|	
11/1/17: 2:35	Next, I found the full path of a file that uses the sector 5000 on the disk image using 'ffind evidence.dd 598' • /eclipse-java-kepler-SR1-win32-x86_64/eclipse/configuration/org.eclipse.osgi/bundles/110/1/.cp/help/EGit/User_Guide/images/Egit-0.9-getstarted-shared-project.png
11/1/17: 2:37	Lastly I used command 'fls evidence.dd' to compare the files I found in task 1

Initials_____KK_____
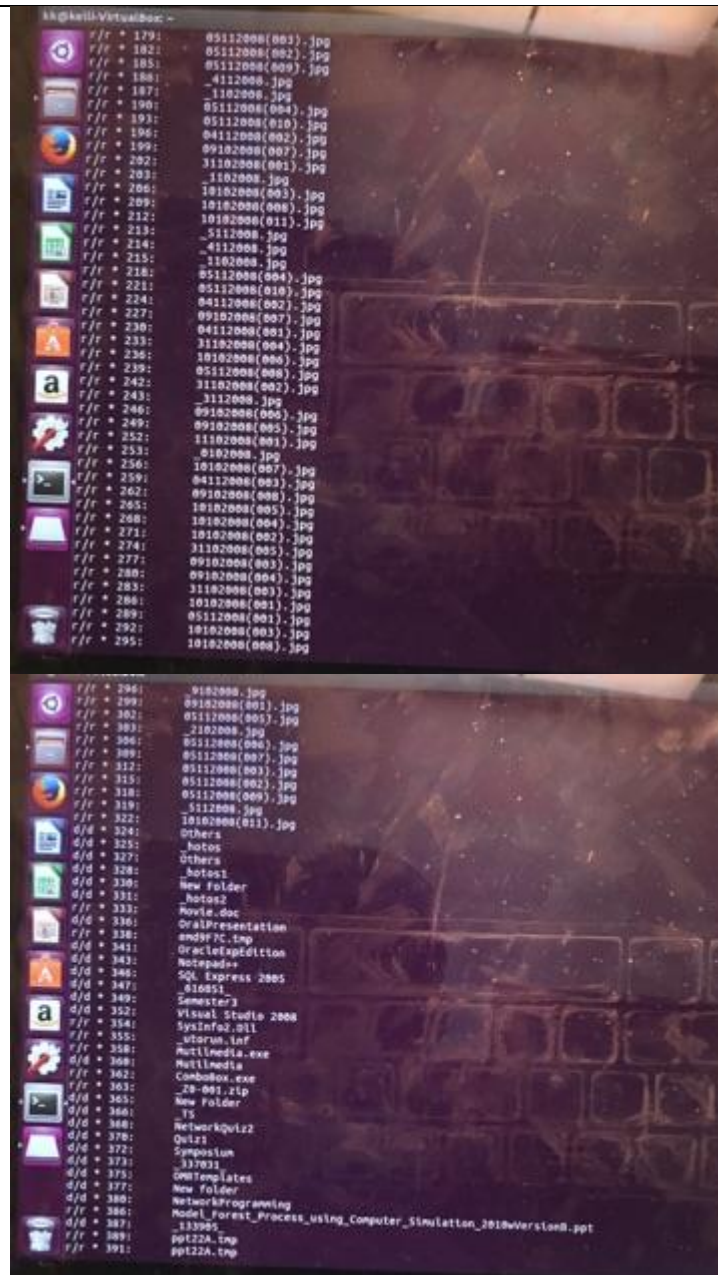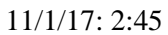
11/1/17: 2:45	I then used 'tsk_recover evidence.dd /home/kk/Desktop/HW_3Drop -e' to find the recovered files for evidence.dd
11/1/17: 2:45	Then had to use 'tsk_recover evidence.dd /home/kk/Desktop/HW_3Drop -a' to find all the recovered files for evidence.dd
11/1/17: 3:00	Lastly, I needed to compare the result with the result in my first task and see if the deleted files were found in my first task. • No, the deleted files were not found