Digital Forensics Tutorials – Acquiring an Image with Kali dcfldd

Explanation Section

Disk Imaging – Definition

Disk images are used to transfer a hard drive's contents for various reasons. A disk image can be used in several instances, including: restoration of a hard drive's contents during disaster recovery, for the transfer of contents of a hard drive from one computer to another, or to restore the contents of a hard drive after hardware upgrade or repair.

A disk Image is defined as a computer file that contains the contents and structure of a data storage device such as a hard drive, CD drive, phone, tablet, RAM, or USB. The disk image consists of the actual contents of the data storage device, as well as the information necessary to replicate the structure and content layout of the device. This differs from a normal backup in that the integrity of the exact storage structure remains intact, which is pivotal in maintaining the integrity of a forensic investigation.

Creating a disk image file of a target is the first step of any digital forensic investigation. In any investigation, analysis is not done on the original data storage device (target), but instead on the exact copy taken.

dd in Kali Linux

dd (disk dump) is a Unix command that is used for a multitude of digital forensic tasks, not least of which is providing a simple means of obtaining a raw image of a file, folder, volume or physical drive. This is essentially the equivalent of creating disk image files in FTK Imager or DiskExplorer for NTFS in Windows. However, this is completed via the terminal with commands in Linux.

dcfldd in Kali Linux

dcfldd is an enhanced version of dd with features useful for forensics and security including:

- On-the-fly hashing – hashing input data as it is being transferred, helping to ensure data integrity.
- Status output - dcfldd can update user on amount of data transferred and time to completion
- Image/wipe & Verify - dcfldd can verify that a target drive is a bit-for-bit match of the specified input file or pattern.
- Multiple outputs - dcfldd can output to multiple files or disks at the same time.
- Split output - dcfldd can split large disk images into multiple files more efficiently than the split command.
- Log output - dcfldd can send all its log data (hash data) and output to text files for easy reading.

dc3dd in Kali Linux

dc3dd is very similar to dcfldd. The largest difference is that dc3dd is based on a slightly different code base. It is a patch, which means it is updated every time dd is updated, whereas dcfldd has its own update and release schedule. dcfldd is preferred by many professionals due to its advanced hashing algorithms and its greater control in how hashing is displayed. However, besides these small differences, both dc3dd and dcfldd have largely the same features.

In This Tutorial

Since dc3dd and dcfldd are so similar in their commands and features, we will be focusing on dcfldd due to its more advanced capabilities in terms of hashing and log output. In this tutorial, we will not only use dcfldd to create a disk image, but will also go through the steps of creating a mount directory, mounting a partition, and writing a text document to the directory. This will create an understanding not only of how to create the disk image, but also of where the disk image data is and how it is created.

Tutorial Section

LEARNING OBJECTIVES:

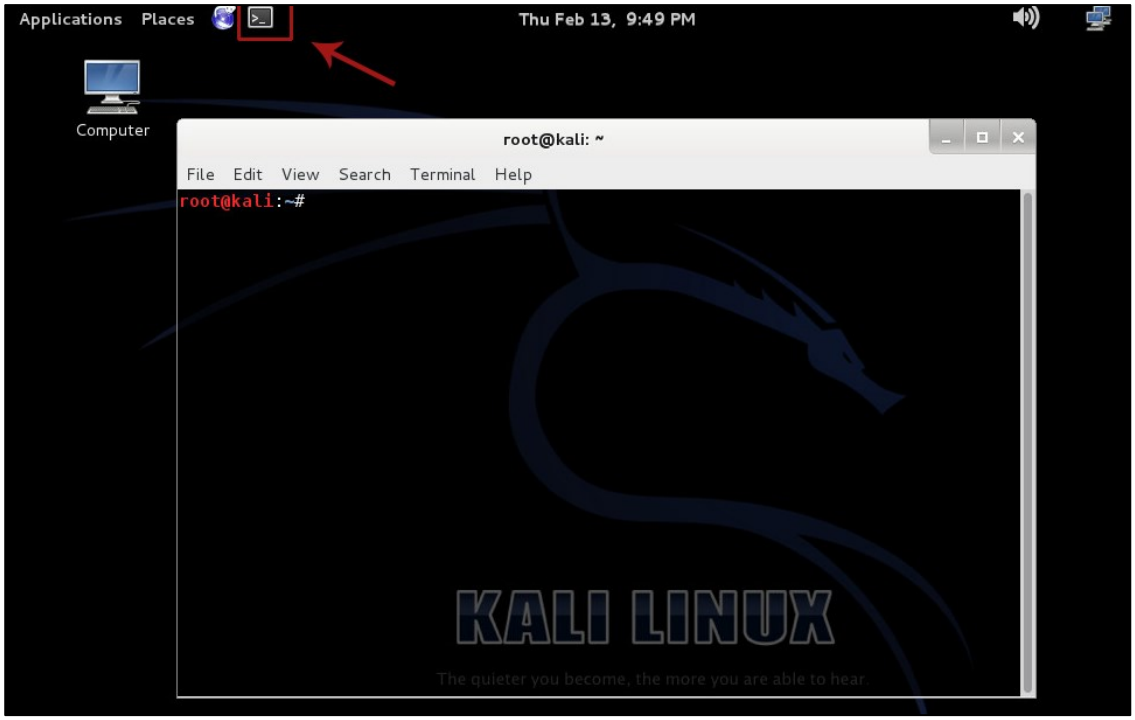
- Identify the partitions available on the system
- Make a mount directory in Kali Linux
- Mount a partition in order to make changes and write data to it
- Navigate to the correct area within the partition to create a new file
- Create a new text file and write data to it
- Create a disk image containing the data written to the partition
- View the hashes created an verify integrity of the disk image

Part 1 – Viewing Kali Linux Partitions

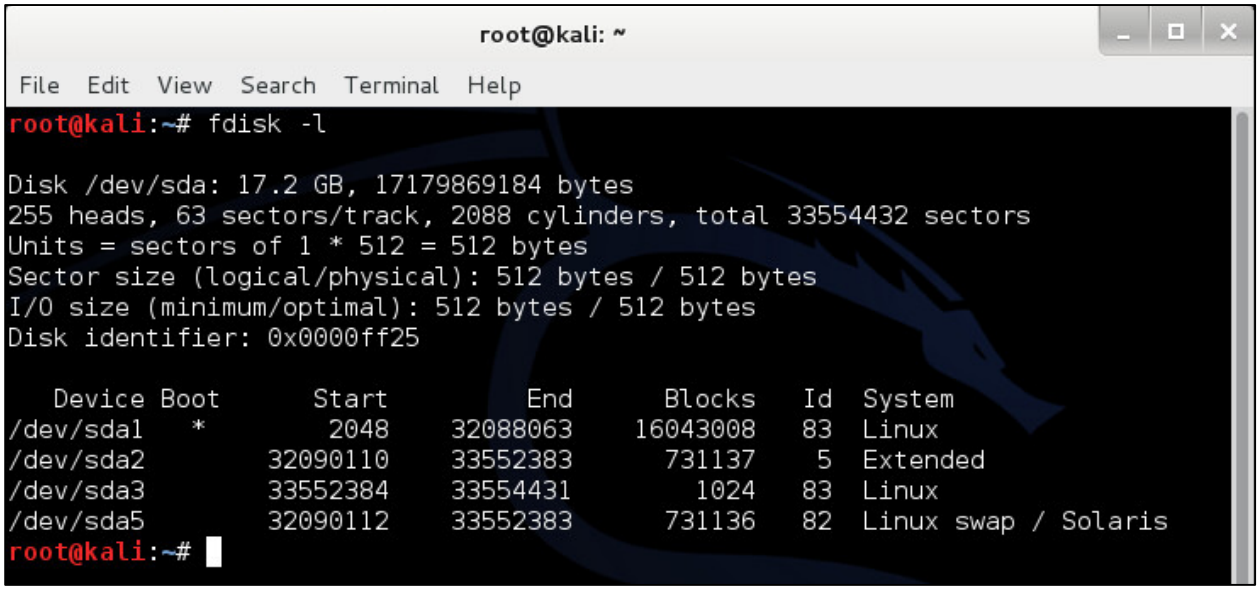
1. Login to the Virtual Lab website (https://v5.unm.edu/cloud/org/ialab), and enter the 'NEST Digital Forensics' vApp. Click on the **Kali Linux machine** to open the VM.

2. At the login screen of the Kali Linux machine use the username **root** and the password **letmein**.

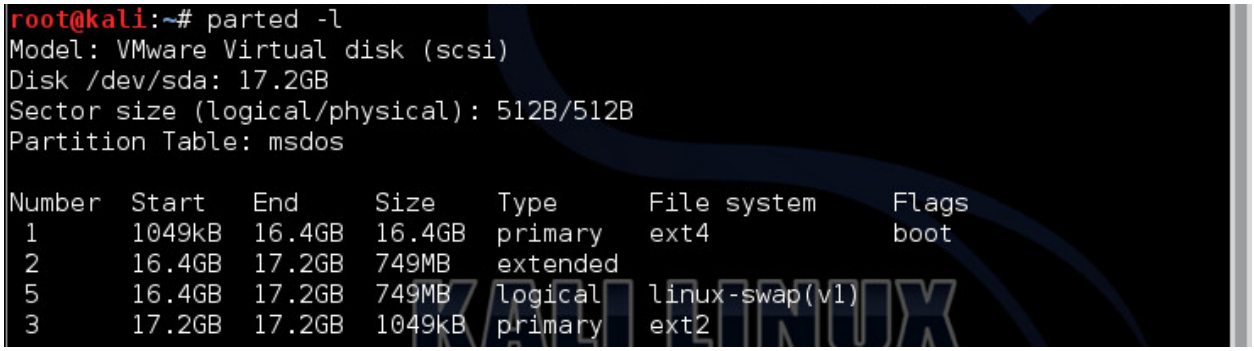
3. Open the Linux terminal. The icon is near the upper left of the screen.



4. Once the terminal has been opened, type the command **fdisk -l**. This will print out a list of the partitions on the Kali machine. You can see that there are four partitions here: sda1, sda2, sda3, and sda5. We will be working with sda3.



5. In order to see the amount of space on each partition, use the command **parted -l**. From this you can see that there is about 1Mb of space on sda3 (the number 3 drive). You can also see that the file system type is **ext2**, which is a default Linux file system type. Also note that this is a primary drive. Linux allows four primary partitions.



Part 2 – Creating a Mount Point & Mounting a Drive

1. sda3 is going to be our "suspect's" drive. We will be creating a disk image of this drive to analyze, just as we did in Windows. However, at this point there is nothing on it. You are going to create a text file on the drive, which will then be copied when the disk image is created.

   In order to write something onto the sda3 drive, the drive first needs to be mounted, like loading a CD or putting in a USB. It needs to be read by the system, and mounting the drive gives it a location that can be seen and accessed.

In order to mount a drive, you must create a location in which to mount it. Just like needing a CD drive to insert a CD, you need a mount location into which to "insert" sda3.

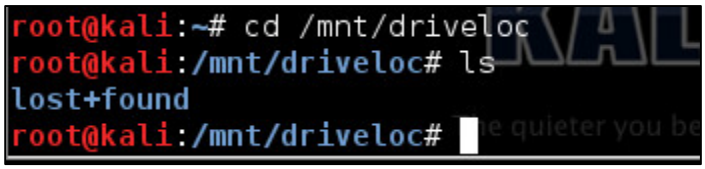
Type **sudo mkdir /mnt/driveloc**. This will create the location /mnt/driveloc.

- **sudo** is equivalent to administrative privileges in Windows. It means "super user does".
- **mkdir** is "make directory".
- **/mnt/driveloc** is simply the folder location you are creating. So now you have a folder called "mnt", and within that a folder called "driveloc".

2. Now that the location has been created, you need to mount sda3 to that location. This will let you see the contents of sda3. Use the command **sudo mount /dev/sda3 /mnt/driveloc** to mount the sda3 drive to the folder 'driveloc'.

Part 3 – Writing a Text File to a Linux Drive

1. Now the drive has been mounted and we can see the contents of the drive. Of course, at this point there is nothing on it. To navigate to the contents of sda3, use the command **cd /mnt/driveloc**. This navigates you to that location. Then type **ls** to see the individual contents of the drive. At this point nothing will appear when using the **ls** command.

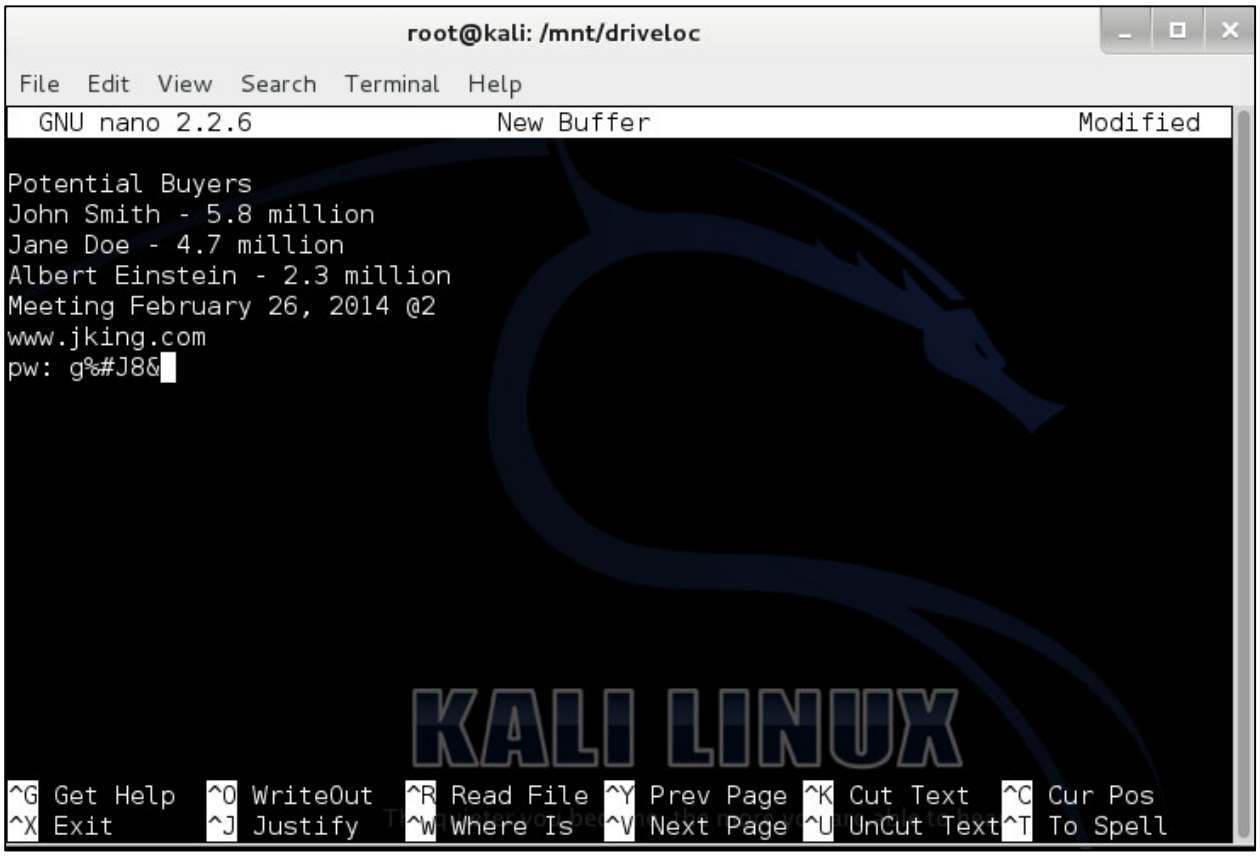


2. We want to write a "suspicious" text file to the sda3 drive. This will then appear when the disk image has been created – if there were photos, email, or other files on this drive, we would see them all upon analyzing the disk image.

   Type **sudo nano**. This will open a text editor in which you can create a new text file. Type some information into the file. Below is some sample text, which has also been placed in the Home Folder on the Kali Linux system if you would like to copy it.

   Potential Buyers
   John Smith – 5.8 million
   Jane Doe – 4.7 million
   Albert Einstein – 2.3 million
   Meeting February 26, 2014 @2
   www.jking.com
   pw: g%#J8&

This text implies that there will be some sort of meeting at a certain website at a certain time for potential buyers of company secrets, and that there is a password to enter the online meeting.



3. Save the file by pressing **CTRL + x**. You will be prompted to write the file. Type **Y** to save the file.

4. You will then be prompted to name the file. Type a name – oftentimes suspects will use misleading names in order to attempt to hide sensitive or suspicious information. I have used **vacationinfo.txt**. After typing the name with the .txt extension, hit **Enter**. This will save the file to /mnt/driveloc.

5. Now if you use the command **ls** you can see that vacationinfo.txt is now present on the drive.

6. Restart the Kali machine in order to unmount the sda3 drive. During the imaging process the target drive should not be mounted in order to avoid accidental changes to the drive.

Part 4 – Imaging the Drive

1. Now that we have mounted the drive and written content to it, we can create a disk image for analysis. Using dcfldd allows on-the-fly hashing. We will be using MD5 and SHA256. SHA256 is a more secure version of SHA1. The fact that it is 256-bit instead of 160-bit decreases the odds of hash collisions (two files generating the same hash).

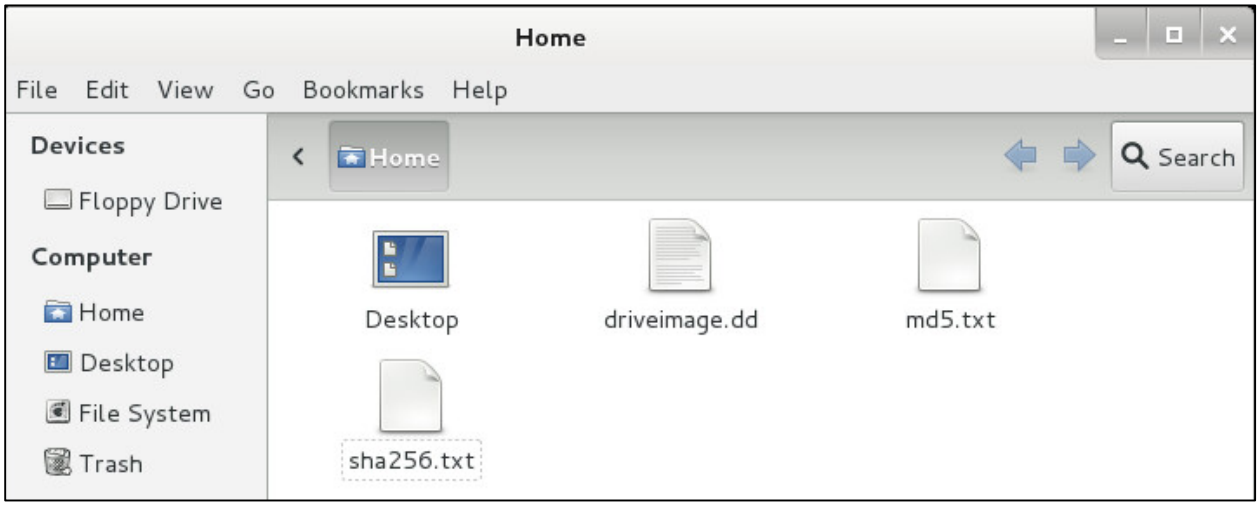
2. Open the terminal and use the following command to create the disk image.
dcfldd if=/dev/sda3 hash=md5,sha256 hashwindow=1G md5log=/root/md5.txt sha256log=/root/sha256.txt hashconv=after conv=noerror,sync of=/root/driveimage.dd

- The command begins with dcfldd. **if=/dev/sda3** designates the drive that will be copied in the disk image.
- **hash=md5,sha256** designates the types of hashes to be generated as the disk image is created.
- **hashwindow=1G** designates the amount of data to be read and copied into the disk image file.
- **md5log=/root/md5.txt sha256log=/root/sha256.txt** designates that the hash logs are going to be printed to two text files within the root folder; that is, within the Home Folder.
- **hashconv=after** designates that the hash values will be written after the disk conversion. Alternatively, 'before' can be selected for this command.
- **conv=noerror,sync** means that if there are read errors when creating the disk image, dcfldd will write zeroes.
- **of=/root/driveimage.dd** is the disk image file that will be created. /root ensures that it will be written to the Home Folder.

*Note: All of these options and more can be seen by navigation to **Applications>>Forensics>>Forensic Imaging Tools>>dcfldd**. This shows the full range of the various dcfldd features.

3. Once the disk image has been completed, the terminal will appear as shown below. dcfldd will also print out how many records were copied during the creation of the disk image.

4. The hash logs and disk image file should now be in the Home Folder. If you open the hash logs you can view the hashes of the image disk that was created.



5. The newly created disk image file must now be set to read-only to make sure that the contents are not altered during analysis. Type **chmod a-w /root/driveimage.dd** in order to change the permissions to read-only.

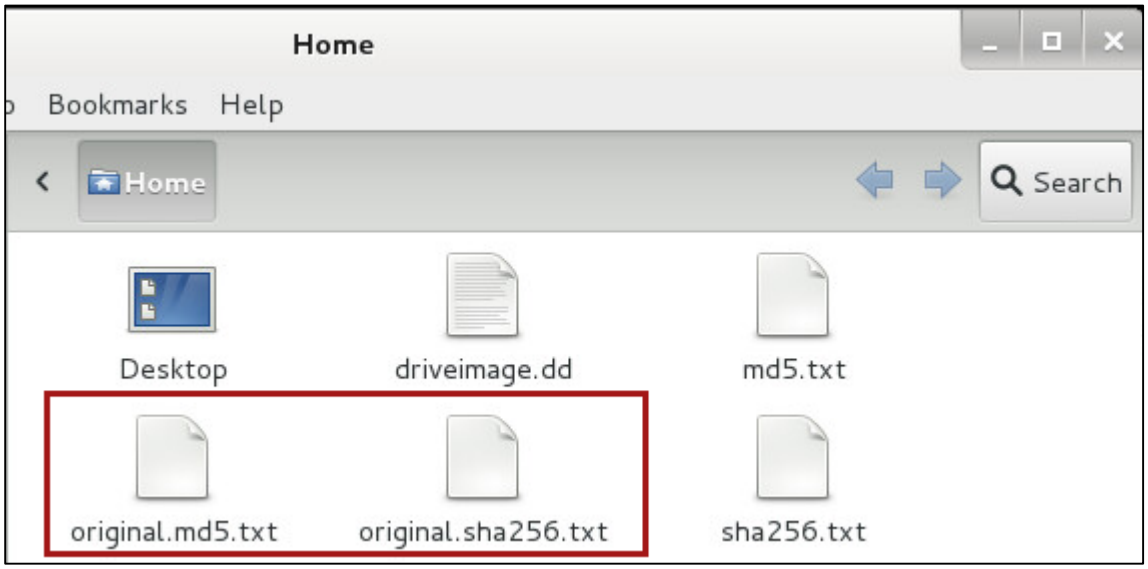
```
root@kali:~# chmod a-w /root/driveimage.dd
```

6. In order to ensure that the original sda3 drive has not been altered, it is necessary to hash the original drive and compare it against the md5 and sha256 hash logs created during the disk image creation process.

   Use the commands **sha256sum /dev/sda3 > /root/original.sha256.txt** and **md5sum /dev/sda3 > /root/original.md5.txt** to create hash files of the original drive. The new text hash files have been placed within the same folder as those created during the imaging process.

```
root@kali:~# sha256sum /dev/sda3 > /root/original.sha256.txt
root@kali:~# md5sum /dev/sda3 > /root/original.md5.txt
root@kali:~#
```

7. Open **Places>>Home Folder** to view the hash files for the original drive. Compare them to md5.txt and sha256.txt hashes and verify that they are identical.



Conclusion

At this point the disk image has been created, write blocked (changed to read-only), and the original drive has been hashed for comparison to make sure no changes have been made. These are exactly the same steps followed in creating a disk image file in Windows, save for the fact that this is done from the terminal in Linux. The disk image is now ready to be analyzed. This will be seen in the following tutorial.

dcfldd is a highly functional imaging tool on Kali Linux. Though this tutorial worked with a very small 1Mb drive, dcfldd is capable of working with enormous drives, which is much more commonly seen in real-world investigation. dcfldd tools allow disk images to be split into multiple files, and to be copied

simultaneously to multiple files and disks. For larger drives this is a highly useful tool that allows for more than simple bit-by-bit copying.