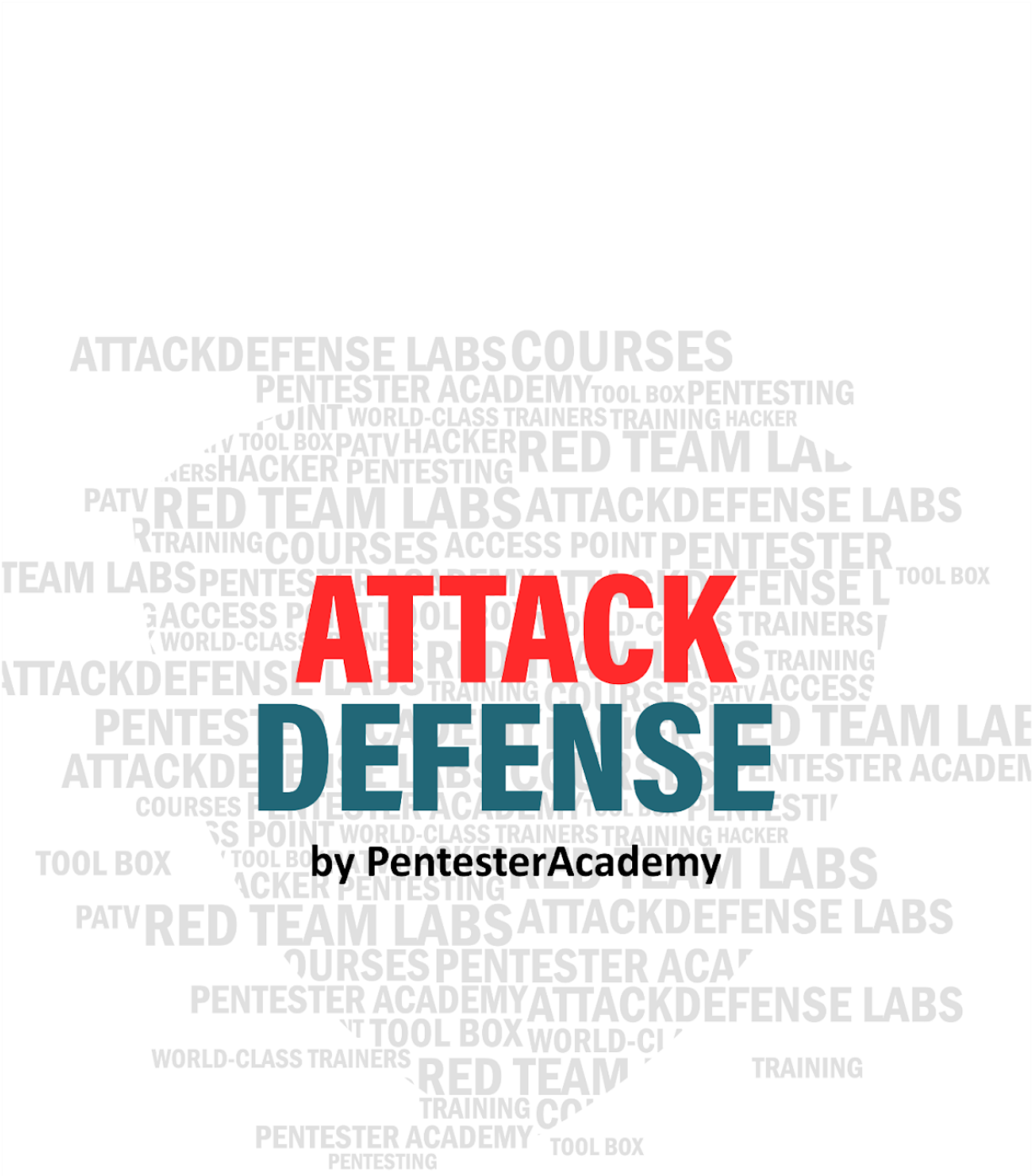
ATTACK
DEFENSE
by PentesterAcademy

Name	Image Acquisition (DD Tools)
URL	https://www.attackdefense.com/challengedetails?cid=1794
Type	Forensics: Disk Forensics

Important Note: This document illustrates all the important steps required to complete this lab. This is by no means a comprehensive step-by-step solution for this exercise. This is only provided as a reference to various commands needed to complete this exercise and for your further research on this topic. Also, note that the IP addresses and domain names might be different in your lab.

Image acquisition involves making a copy (or several copies) of the seized hard disk which can be then used to forensics analysis. This allows the investigators to analyze this image while ensuring the integrity and present condition of the real evidence disk.

In this lab, the evidence hard disk is mounted on '/dev/sdc'. The dd tools are installed on the lab machine.

Objective: Create a disk image for evidence hard disk using dd tools.

Solution:

Step 1: Verify that the evidence hard drive is mounted.

Command: df -h

```
root@localhost:~# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/root       2.0G  1.6G  211M  89% /
devtmpfs        1.5G     0  1.5G   0% /dev
tmpfs           1.5G     0  1.5G   0% /dev/shm
tmpfs           1.5G  448K  1.5G   1% /run
tmpfs           5.0M     0  5.0M   0% /run/lock
tmpfs           1.5G     0  1.5G   0% /sys/fs/cgroup
/dev/sdb        976M  2.6M  907M   1% /root
/dev/sdc        240M   95M  129M  43% /mnt/evidence
tmpfs           300M     0  300M   0% /run/user/0
```

The evidence disk is mounted at /mnt/evidence directory.

Step 2: It is recommended to unmount the disk for preventing any failures during disk imaging. Unmount and verify that the disk is unmounted.

Commands:
umount /mnt/evidence
df -h

```
root@localhost:~# umount /mnt/evidence
root@localhost:~# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/root       2.0G  1.6G  211M  89% /
devtmpfs        1.5G     0  1.5G   0% /dev
tmpfs           1.5G     0  1.5G   0% /dev/shm
tmpfs           1.5G  448K  1.5G   1% /run
tmpfs           5.0M     0  5.0M   0% /run/lock
tmpfs           1.5G     0  1.5G   0% /sys/fs/cgroup
/dev/sdb        976M  2.6M  907M   1% /root
tmpfs           300M     0  300M   0% /run/user/0
root@localhost:~#
```

Step 3: Use the dd utility to create a disk image. Pass the full path of the evidence drive and output image name as an argument.

Command: dd if=/dev/sdc of=evidence.img

```
root@localhost:~# dd if=/dev/sdc of=evidence.img
524288+0 records in
524288+0 records out
268435456 bytes (268 MB, 256 MiB) copied, 26.9729 s, 10.0 MB/s
root@localhost:~#
```

The evidence.img file is created successfully by using dd utility.

Step 4: Check the disk image which has been created.

Commands:
ls -l
file evidence.img

```
root@localhost:~# ls -l
total 262144
-rw-r--r-- 1 root root 268435456 Apr 19 05:10 evidence.img
root@localhost:~# file evidence.img
evidence.img: Linux rev 1.0 ext4 filesystem data, UUID=6b832003-6bf1-4786-a3c1-e252e8f99090 (extents) (64bit) (large files)
 (huge files)
root@localhost:~#
```

Step 5: Create the md5sum of the disk image.

Command: md5sum evidence.img

```
root@localhost:~#
root@localhost:~# md5sum evidence.img
4a2950477e0d5c097fb57b511a5638d6  evidence.img
root@localhost:~#
```

Note: The md5sum hash will be different for every instance of the lab machine. After getting the hash for the first time, it should be the same throughout that lab instance for all the tools.

Alternate Approach: Using dcfldd

dcfldd is a modified version of GNU originally created by Nicholas Harbour from the DoD Computer Forensics Laboratory (DCFL). It supports hashing, fast disk wiping (through patterns) and status output.

Step 1: Use dcfldd utility to create a disk image. Pass the full path of the evidence drive, output image name, hash type, log file as arguments.

Name: evidence.img
Hash: md5 and sha256
Log: sha.log and md5.log

Command: dcfldd if=/dev/sdc hash=md5,sha256 md5log=md5.log sha256log=sha.log of=evidence.img

```
root@localhost:~# dcfldd if=/dev/sdc hash=md5,sha256 md5log=md5.log sha256log=sha.log of=evidence.img
8192 blocks (256Mb) written.
8192+0 records in
8192+0 records out
root@localhost:~#
```

The evidence.img file is created with a log file.

Step 2: Check the disk image which has been created.

Commands:
ls -l
file evidence.img

```
root@localhost:~# ls -l
total 262152
-rw-r--r-- 1 root root 268435456 Apr 19 05:13 evidence.img
-rw-r--r-- 1 root root        47 Apr 19 05:13 md5.log
-rw-r--r-- 1 root root        82 Apr 19 05:13 sha.log
root@localhost:~#
root@localhost:~# file evidence.img
evidence.img: Linux rev 1.0 ext4 filesystem data, UUID=6b832003-6bf1-4786-a3c1-e252e8f99090 (extents) (64bit) (large files)
 (huge files)
root@localhost:~#
```

The md5sum of the image is automatically calculated and stored in md5.log

Step 3: Check the MD5 stored in md5.log

Command: cat md5.log

```
root@localhost:~# cat md5.log

Total (md5): 4a2950477e0d5c097fb57b511a5638d6
root@localhost:~#
```

Splitting the Disk Image

This approach can help while imaging a large evidence disk. The smaller parts of the image can be then sent over the internet or carried on relatively smaller portable storage devices.

Step 1: Use dcfldd utility to create a disk image in multiple parts. Pass the full path of the evidence drive, output image name, hash type, log file, split size, split format as arguments.

Name: evidence.img
Hash: md5 and sha256
Log: sha.log and md5.log
Split Size: 64M
Split Format: 000

Command: dcfldd if=/dev/sdc hash=md5,sha256 md5log=md5.log split=64M splitformat=000 sha256log=sha.log of=evidence.img

```
root@localhost:~# dcfldd if=/dev/sdc hash=md5,sha256 md5log=md5.log split=64M splitformat=000 sha256log=sha.l
og of=evidence.img
8192 blocks (256Mb) written.
8192+0 records in
8192+0 records out
root@localhost:~#
```

Step 2: Check the disk image parts created by the tool.

Command: ls -l

```
root@localhost:~# ls -l
total 262152
-rw-r--r-- 1 root root 67108864 Apr 19 05:15 evidence.img.000
-rw-r--r-- 1 root root 67108864 Apr 19 05:15 evidence.img.001
-rw-r--r-- 1 root root 67108864 Apr 19 05:15 evidence.img.002
-rw-r--r-- 1 root root 67108864 Apr 19 05:15 evidence.img.003
-rw-r--r-- 1 root root       47 Apr 19 05:15 md5.log
-rw-r--r-- 1 root root       82 Apr 19 05:15 sha.log
root@localhost:~#
```

The 256 MB image is divided into 4 parts. These can be now transported.

Step 3: Check the md5.log for the MD5 hash.

```
root@localhost:~# cat md5.log

Total (md5): 4a2950477e0d5c097fb57b511a5638d6
root@localhost:~#
```

Step 4: On the destination, the analyst will need the original image. So, combine the splitted disk images into ‘evidence.img’ image.

Commands:
cat evidence.img.00* > evidence.img
ls -l

```
root@localhost:~# cat evidence.img.00* > evidence.img
root@localhost:~# ls -l
total 524296
-rw-r--r-- 1 root root 268435456 Apr 19 05:16 evidence.img
-rw-r--r-- 1 root root  67108864 Apr 19 05:15 evidence.img.000
-rw-r--r-- 1 root root  67108864 Apr 19 05:15 evidence.img.001
-rw-r--r-- 1 root root  67108864 Apr 19 05:15 evidence.img.002
-rw-r--r-- 1 root root  67108864 Apr 19 05:15 evidence.img.003
-rw-r--r-- 1 root root        47 Apr 19 05:15 md5.log
-rw-r--r-- 1 root root        82 Apr 19 05:15 sha.log
root@localhost:~#
```

Step 5: Calculate the MD5 hash for the combined image using md5sum utility.

Command: md5sum evidence.img

```
root@localhost:~#
root@localhost:~# md5sum evidence.img
4a2950477e0d5c097fb57b511a5638d6  evidence.img
root@localhost:~#
```

The calculated MD5 and the MD5 sum generated by the dcfldd tool (present in md5.log file) is the same. Hence, the disk is combined properly.

References:

1. dc3dd (https://sourceforge.net/projects/dc3dd/)
2. dcfldd (https://github.com/adulau/dcfldd)