Module 3

LINUX File System

- A Linux file system is a structured collection of files on a disk drive or a partition.
- It is generally a built-in layer of a Linux operating system used to handle the data management of the storage.
- It helps to arrange the file on the disk storage.
- It manages the file name, file size, creation date, and much more information about a file.

- The Linux file system contains the following sections:
  - The root directory (/)
  - A specific data storage format (EXT3, EXT4, BTRFS, XFS and so on)
  - A partition or logical volume having a particular file system.

Linux File System Structure

- Linux file system has a hierarchal file structure as it contains a root directory and its subdirectories.
- All other directories can be accessed from the root directory.
- A partition usually has only one file system, but it may have more than one file system.

- In Linux, the file system creates a tree structure. All the files are arranged as a tree and its branches. The topmost directory called the **root (/) directory**. All other directories in Linux can be accessed from the root directory.

Table 7-1. Linux System Files

System file	Contents
/etc/exports	File systems exported to remote hosts; might include remote drive mappings
/etc/fstab	File system table of devices and mount points
/var/log/lastlog	User's last logon
/var/log/wtmp	Logon and logoff history information
/var/run/utmp	Current user's logon information
/var/log/dmesg	System messages log
/var/log/syslog	System log, occasionally called system.log or kernel.log
/etc/shadow	Master password file, containing hashed passwords for the local system
/etc/group	Group memberships for the local system
/etc/passwd	Account information for the local system

Table 7-2. Core Top-Level Directories of a Linux System

Directory	Contents
/usr	Most applications and commands are in this directory or its subdirectories bin (stands for "binary" and contains binary files required at boot time) and sbin (which requires superuser permission to run the binaries in it).
/etc	Most system configuration files are stored in this directory.
/home	The home directories for all users, usually named after their usernames.
/root	The home directory for the root user (superuser), which is kept separate from other user home directories.
/dev	Device files that act as stand-ins for the devices they represent, as described in Chapter 3; for example, /dev/sda is the first non-IDE disk drive on the system, usually the main hard drive.
/var	Subdirectories such as log (often useful for investigations), mail (storing e-mail accounts), and spool (where print jobs are spooled).

File Structures in Ext4

- Linux supports a wide range of file systems.
- The early standard was Second Extended File System (Ext2) , and then Third Extended File System (Ext3) replaced Ext2 in most Linux distributions.
- Its major difference from Ext2 was being a journaling file system, which has a built-in file recovery mechanism used after a crash.

Fourth Extended File System (Ext4)

- Among other features, it added support for partitions larger than 16 TB, improved management of large files, and offered a more flexible approach to adding file system features.
- Because these changes affected the way the Linux kernel interacts with the file system, adoption of Ext4 was slower in some Linux distributions, but it’s now considered the standard file system for most distributions.
- The Ubuntu version you used previously, for example, has an Ext4 partition at its core, unless you select another file system during installation.

- In UNIX and Linux, everything is considered a file, including disk drives, monitors, tape drives, network interface cards, system memory, and directories.

- UNIX/Linux has four components defining the file system:
- boot block,
- superblock,
- inode block
- data block.

- A block is the smallest disk allocation unit in the UNIX/Linux file system and can be 512 bytes and up; block size depends on how the disk volume is initiated.

- The boot block contains the bootstrap code—instructions for startup. A UNIX/Linux computer has only one boot block, on the main hard disk.

- The superblock contains vital information about the system and is considered part of the metadata.
- It specifies the disk geometry and available space and keeps track of all inodes.
- The superblock also manages the file system, including configuration information, such as block size for the drive, file system names, blocks reserved for inodes, and volume name.
- Multiple copies of the superblock are kept in different locations on the disk to prevent losing such important information

- Inode blocks contain the first data after the superblock. An inode is assigned to every file allocation unit.
- As files or directories are created or deleted, inodes are also created or deleted.
- The link between inodes associated with files and directories controls access to those files or directories.

- The data block is where directories and files are stored on a disk drive.
- This location is linked directly to inodes.
- As in Microsoft file systems, the Linux file system on a PC has 512-byte sectors.
- A data block is equivalent to a cluster of disk sectors on a FAT or NTFS volume. Blocks range from 1024 to 4096 bytes each on a Linux volume.

INODES

- Inodes contain file and directory metadata and provide a mechanism for linking data stored in data blocks.
- When a file or directory is created on a Linux file system, an inode is assigned.

inode contains the following information:

- The mode and type of the file or directory
- The number of links to a file or directory
- The UID and GID of the file's or directory's owner
- The number of bytes in the file or directory
- The file's or directory's last access time and last modified time
- The inode's last file status change time(When the inode itself (like permissions or ownership) was last changed.)

- The block address for the file data
- The indirect, double-indirect, and triple-indirect block addresses for the file data
- Current usage status of the inode
- The number of actual blocks assigned to a file
- File generation number and version number
- The continuation inode’s link

- An assigned inode has 13 pointers that link to data blocks and other pointers where files are stored.
- Pointers 1 through 10 link directly to data storage blocks in the disk's data block and contain block addresses indicating where data is stored on the disk.
- These pointers are direct pointers because each one is associated with one block of data storage.

Inode Pointers :

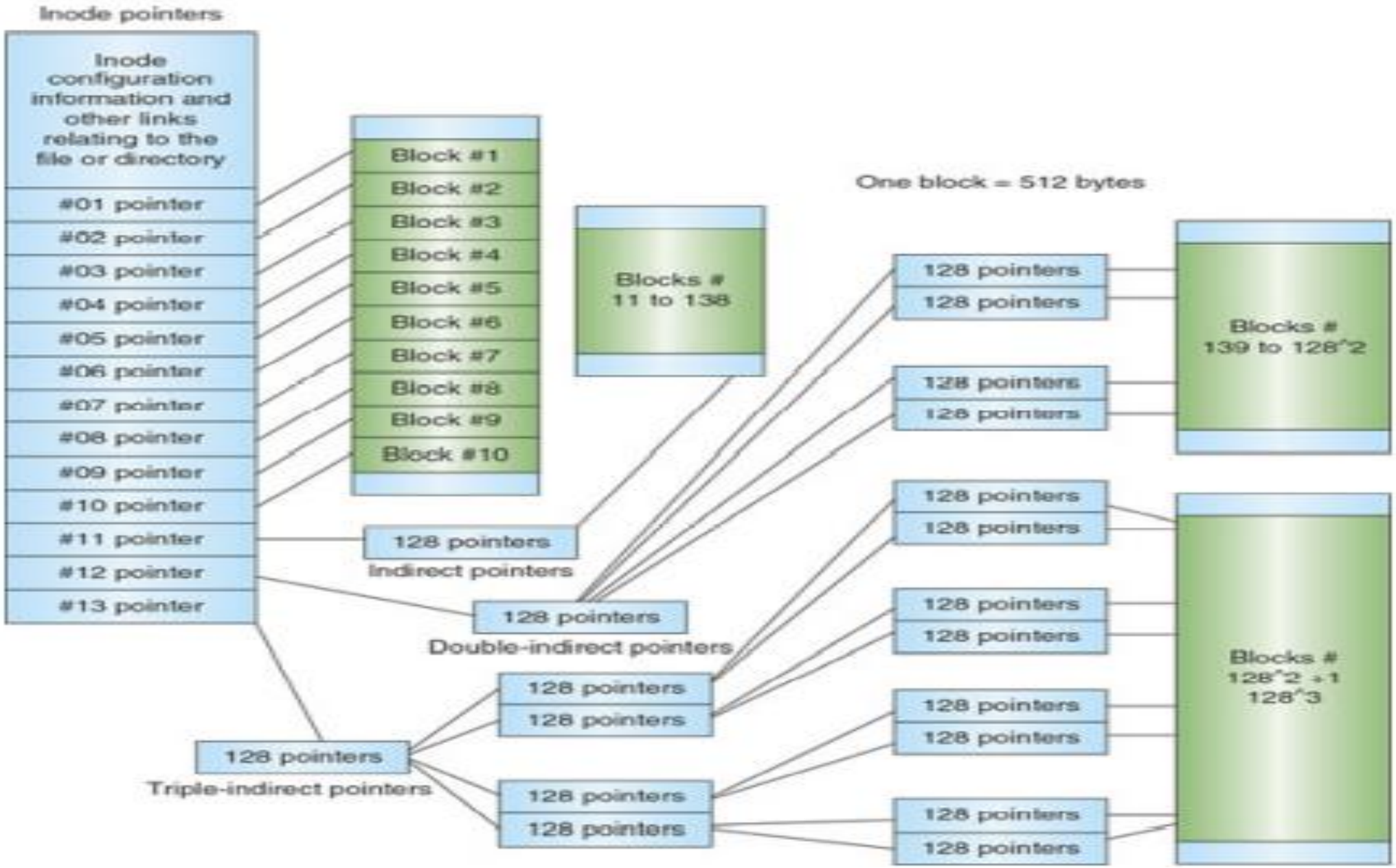
1.Direct Pointers (#1–10): These pointers link directly to data blocks where the file data is stored. Each pointer corresponds to one block of data (with block size shown as 512 bytes in the image).
2.Indirect Pointer (#11): Points to a block that holds more pointers. Each of these pointers further points to a data block. This layer is used when a file exceeds what can be stored using direct pointers alone.
3.Double-Indirect Pointer (#12): Points to a block of pointers. These pointers themselves lead to other blocks containing even more pointers, which finally point to data blocks.
4.Triple-Indirect Pointer (#13): Links to a block of pointers that point to other blocks of pointers, which in turn link to blocks of yet more pointers before reaching the actual data blocks.

Layered Structure (Data Access via Pointers):

- **Direct Pointers**: For small files (stored in up to 10 blocks), the filesystem directly accesses data blocks using the first 10 pointers.
- **Indirect Pointers**: For larger files that cannot fit within 10 blocks, the 11th pointer accesses additional data by pointing to a block of 128 pointers.
- **Double-Indirect Pointers**: For even larger files, the 12th pointer refers to blocks that hold pointers to other blocks of pointers, which then link to data blocks.
- **Triple-Indirect Pointers**: For very large files, the 13th pointer adds another layer of indirection, pointing to blocks that contain pointers to other blocks, which eventually point to data blocks.

Example:

- The first block linked by the double-indirect pointer (12th) is Block 139. This represents the transition from direct and single-indirect pointers to a more complex structure for handling larger data sets.
- If more storage is needed, the triple-indirect pointer (13th) introduces an additional layer of references.

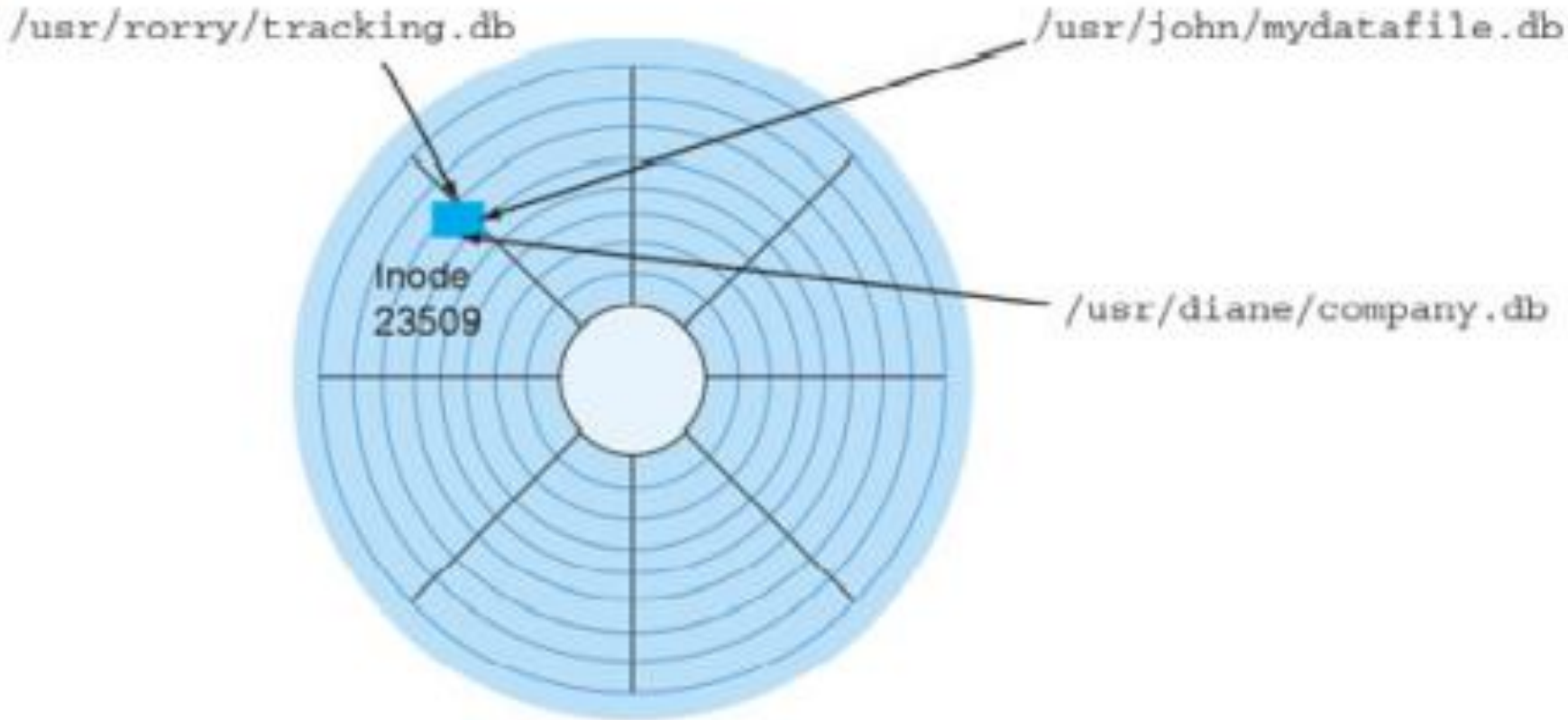

Inode pointers
Inode configuration information and other links relating to the file or directory
#01 pointer
#02 pointer
#03 pointer
#04 pointer
#05 pointer
#06 pointer
#07 pointer
#08 pointer
#09 pointer
#10 pointer
#11 pointer
#12 pointer
#13 pointer
Block #1
Block #2
Block #3
Block #4
Block #5
Block #6
Block #7
Block #8
Block #9
Block #10
Blocks # 11 to 138
One block = 512 bytes
128 pointers
Indirect pointers
128 pointers
Double-indirect pointers
128 pointers
Triple-indirect pointers
128 pointers
128 pointers
Blocks # 139 to 128^2
Blocks # 128^2 +1 128^3

Hard Links and Symbolic Links

- A hard link is a pointer that allows accessing the same file by different filenames
- The filenames refer to the same inode and physical location on a drive.
- Originally, hard links were used so that people with different logins could access the same physical file.
- If one person changed the file, the changes would be apparent when another user opened the file

- Figure shows three hard-linked files pointing to the same inode: 23509.
- You use the ln command to create a hard link.
- The main requirement is that all files pointing to the same inode have to be on the same physical drive, not on another volume.

Figure 7-4.



Hard-linked files with different filenames

- To see files and their inode numbers, you use the ls -ia command. Inside each inode is a field called link count that specifies the number of hard links.
- The link count for directories is higher than for other file types. If two files have the same inode number, the link count is two. If one file is deleted, the link count drops by one.
- When the hard link count drops to zero, the file is effectively deleted. Most forensics tools, however, can retrieve these files.
- To see the contents of a directory, you use the ls -a command.

Symbolic links

- also known as “soft links” or “symlinks”
- They are simply pointers to other files and aren’t included in the link count.
- Unlike hard links, they can point to items on other drives or other parts of the network; they simply need an absolute path. Symbolic links have an inode of their own, which isn’t the same as the inode of the item they’re pointing to.

- Unlike hard links, they depend on the continued existence of the destination they’re pointing to, and they’re easier to identify on a running Linux system than hard links are.
- Unlike hard links, which point to their destination with an inode number, symbolic links identify their destination by name and path.
- If a name and path no longer exist, the symbolic link stops working. You create symbolic links with the ln -s command.

Example of a Hard Link:

echo "Hello World" > file1.txt
ln file1.txt hardlink1.txt
rm file1.txt
Even though file1.txt is deleted, the file data still exists and can be accessed via hardlink1.txt.

Example of a Soft Link (Symbolic Link):

echo "Hello Soft Link" > file2.txt
ln -s file2.txt softlink1.txt
softlink1.txt is a symbolic link pointing to file2.txt. When you open softlink1.txt, it will lead you to the content of file2.txt.

rm file2.txt
Now, if you try to open softlink1.txt, you'll get an error saying the file doesn't exist because the original file (file2.txt) is deleted, leaving the symbolic link broken.

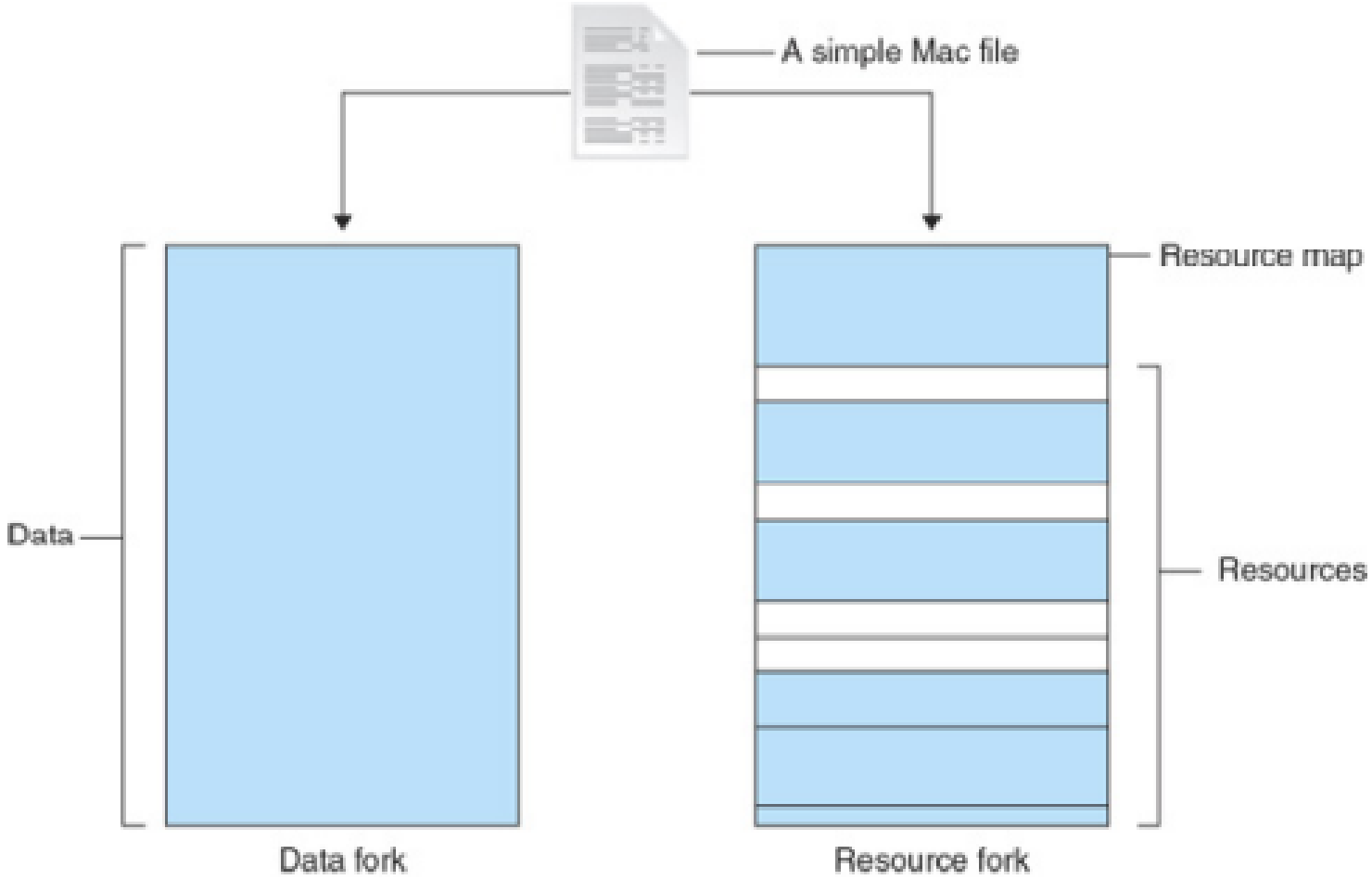
Understanding Macintosh File Structures

- The latest stable version of macOS is 10.15.1 which is also known as Catalina.
- The current major version, macOS 12 Monterey, was released on June 22, 2021
- Other versions still in use include 10.12.5 (Sierra), 10.11 (El Capitan), 10.9 (Yosemite), 10.6 (Snow Leopard), 10.7 (Lion), and 10.8 (Mountain Lion). macOS is built with the new Apple File System (APFS).
- The current version offers better security, encryption, and performance speeds, and users can still mount HFS+ drives (Hierarchical File System)one of the primary file systems of macOS).

An Overview of Mac File Structures

- In macOS, a file consists of two parts: a data fork , where data is stored, and a resource fork , where file metadata and application information are stored.

Figure 7-9.

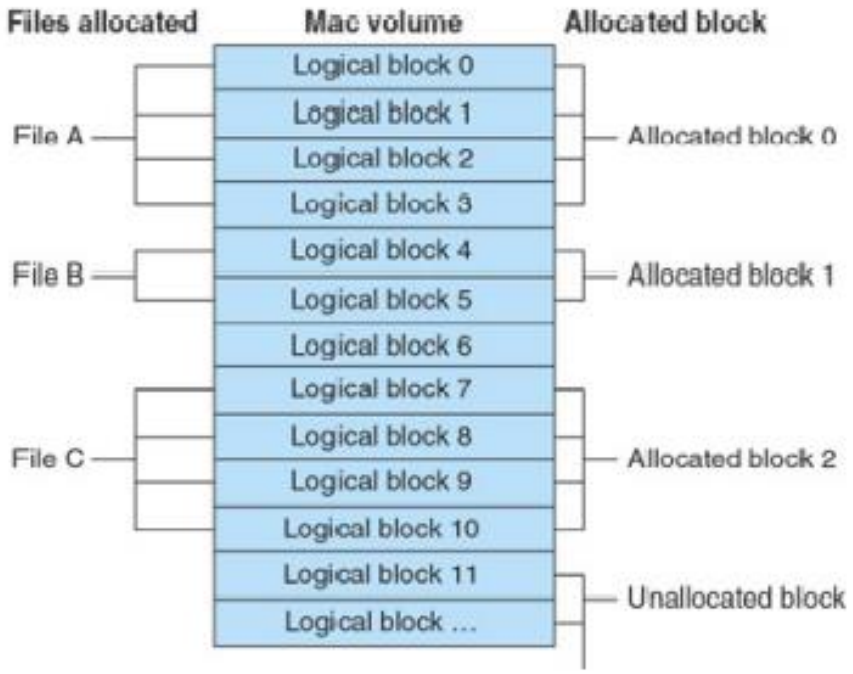


The resource fork and data fork in a macOS file

- The data fork typically contains data the user creates, such as text or spreadsheets.
- Applications, such as Microsoft Word or Excel, also read and write to the data fork.
- When you’re working with an application file, the resource fork contains additional information, such as menus, dialog boxes, icons, executable code, and controls.

- Volumes have allocation blocks and logical blocks.
- A logical block is a collection of data that can’t exceed 512 bytes.
- When you save a file, it’s assigned to an allocation block, which is a group of consecutive logical blocks.
- As volumes increase in size, one allocation block might be composed of three or more logical blocks.

- Figure shows the relationship between these two types of blocks.



Logical and allocation block structures

- If a file contains information, it always occupies one allocation block.
  - For example, if a data fork contains only 11 bytes of data, it occupies one allocation block (512 bytes) on a disk, which leaves more than 500 bytes empty in the data fork.

- In macOS, file fragmentation is reduced by using clumps , which are groups of contiguous allocation blocks.
- As a file increases in size, it occupies more of the clump.
- Volume fragmentation is kept to a minimum by adding more clumps to larger files.

- For older HFS-formatted drives, the first two logical blocks, 0 and 1, on the volume (or disk) are the boot blocks containing system startup instructions.
- Optional executable code for system files can also be placed in boot blocks.

- Older Mac OSs use the Master Directory Block (MDB) for HFS, which is the Volume Information Block (VIB) in HFS+.
- All information about a volume is stored in the MDB and written to it when the volume is initialized.
- A copy of the MDB is also written to the next-to last block on the volume to support disk utility functions.
- When the OS mounts a volume, some information from the MDB is written to a Volume Control Block (VCB) , stored in system memory. When a user no longer needs the volume and unmounts it, the VCB is removed.

- The copy of the MDB is updated when the extents overflow file or catalog increases in size.
- The extents overflow file is used to store any file information, not in the MDB or a VCB.
- The catalog is the listing of all files and directories on the volume and is used to maintain relationships between files and directories on a volume.

Forensic procedures in MacOS

- For forensics procedures in macOS, you must know where file system components are located and how both files and file components are stored.
- In Mac OS ,Application settings are in three formats:
  - Plaintext(can be viewed in any text editor.)
  - plist files (which include plain XML plists and binary plists, which are condensed XML)
  - SQLite database. Plaintext files, of course (To view the SQLite database, use the SQLite Database Browser)

- Plist files are preference files for installed applications on a system, usually stored in /Library/Preferences.
- To view them, you use special editors, such as the one available at the Apple Developer Web site.

New macOS feature - Unified logging

- which is located in /var/db/diagnostics (where log files are stored) and /var/db/uuid.text.
- It includes three new utilities—log, log collect, and log show—that a forensics examiner can use.

Other files that might contain information useful for an investigation include the following:

- /System/Library/CoreServices/SystemVersion.plist—Contains the OS version.
- /Library/Preferences/SystemConfiguration/NetworkInterfaces.plist—Shows all existing network interfaces. If an interface has been used recently, it's listed in the /private/var/db/dhcpclient/leases directory.
- /private/var/db/DirectoryService/flatfile.db—A list of users on a system; used before Mac OS X v10.7 and is similar to the Linux/UNIX /etc/passwd file.
- /private/var/db/dslocal/nodes/Default/users—Contains users' plist files in Mac OS X after v10.7.
- /private/var/db/shadow/hash—Contains account passwords.

- **FileVault,** introduced with version 10.3, is used to encrypt and decrypt a user's /users directory.
- It has master keys and recovery keys, which research later proved could be retrieved from RAM and used to crack encryption.
- In response to these security vulnerabilities, the improved FileVault2 was introduced, which encrypts the whole disk with 128-bit AES encryption.

- Other encrypted information you're likely to find during an investigation is passwords.
- Since Mac OS 8.6, **keychains** have been used to manage passwords for applications, Web sites, and other system files
- You can find keychain files in a variety of places, including /System/Library/Keychains and /Library/Keychains, and they can be useful to show what applications and files require passwords.
- The Mac application Keychain Access enables you to restore passwords.

Vendor softwares used for examining the macOS file system

- BlackBag Technologies
- SubRosaSoft MacForensicsLab
- ProDiscover Forensic Edition
- Sleuth Kit and Autopsy

Acquisition Methods in macOS

- To examine a computer running macOS, you need to make an image of the drive
- Use a macOS-compatible forensic boot CD/DVD to make an image, which then must be written to an external drive, such as a FireWire or USB drive.
- Larger macOS systems are constructed much like desktop PCs, making removal of the hard drive easier.

- BlackBag Technologies sells acquisition tools for OS 9 and OS X and offers a forensic boot CD called MacQuisition for making a drive image.

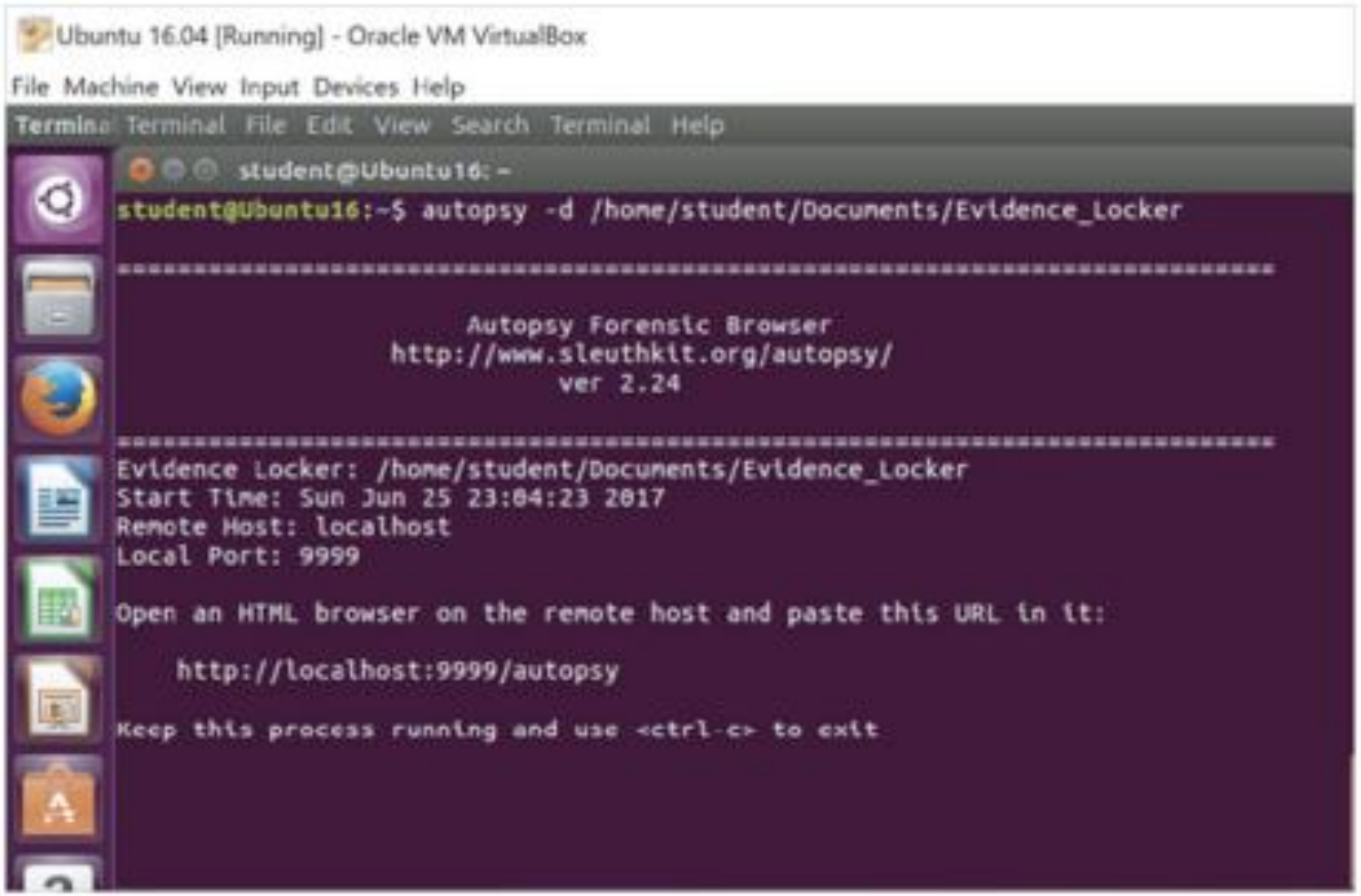
- After making an acquisition, the next step is examining the image of the file system with a forensics tool.
- The tool you use depends on the image file's format.
  - For example, if you used EnCase, FTK, or X-Ways Forensics to create an Expert Witness (.e0l) image, you must use one of these tools to analyze the image.

- If you made a raw format image, you can use any of the following tools:
  - BlackBag Technologies Macintosh Forensic Software (OS X only)
  - SubRosaSoft MacForensicsLab (OS X only)
  - Guidance Software EnCase Recon Mac OS X
  - Forensics with Palladin
  - X-Ways Forensics
  - AccessData FTK

Installing Sleuth Kit and Autopsy

- In Linux, Sleuth Kit must be installed before Autopsy Forensic Browser, or Autopsy isn’t installed correctly.
- In Windows, however, you just need to install Autopsy.
- In addition, when you’re running Autopsy Forensic Browser in Mac or Linux, you must preface all commands with sudo.

- To install Sleuth Kit and Autopsy Forensic Browser in Ubuntu 16.04, you need root user privileges. Follow these steps:
  - If necessary, start Ubuntu and open a terminal window.
  - To install Sleuth Kit, type **sudo apt-get install sleuthkit** and press Enter, and then install Autopsy by typing **sudo apt-get install autopsy** and pressing Enter.
    - To confirm that you're in your home directory, type pwd and press Enter.
  - Next, create the evidence locker for storing files by typing **mkdir Documents/Evidence_Locker** and pressing Enter.
  - To start Autopsy and let it know where to store files, type **autopsy -d /home/username/Documents/Evidence_Locker** and press Enter.


Ubuntu 16.04 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminal File Edit View Search Terminal Help
student@Ubuntu16: ~
student@Ubuntu16:~$ autopsy -d /home/student/Documents/Evidence_Locker
Autopsy Forensic Browser
http://www.sleuthkit.org/autopsy/
ver 2.24
Evidence Locker: /home/student/Documents/Evidence_Locker
Start Time: Sun Jun 25 23:04:23 2017
Remote Host: localhost
Local Port: 9999
Open an HTML browser on the remote host and paste this URL in it:
http://localhost:9999/autopsy
Keep this process running and use <ctrl-c> to exit

- Right-click the URL http://localhost:9999/autopsy shown in the terminal window and click Open Link. Figure shows the Autopsy main window.

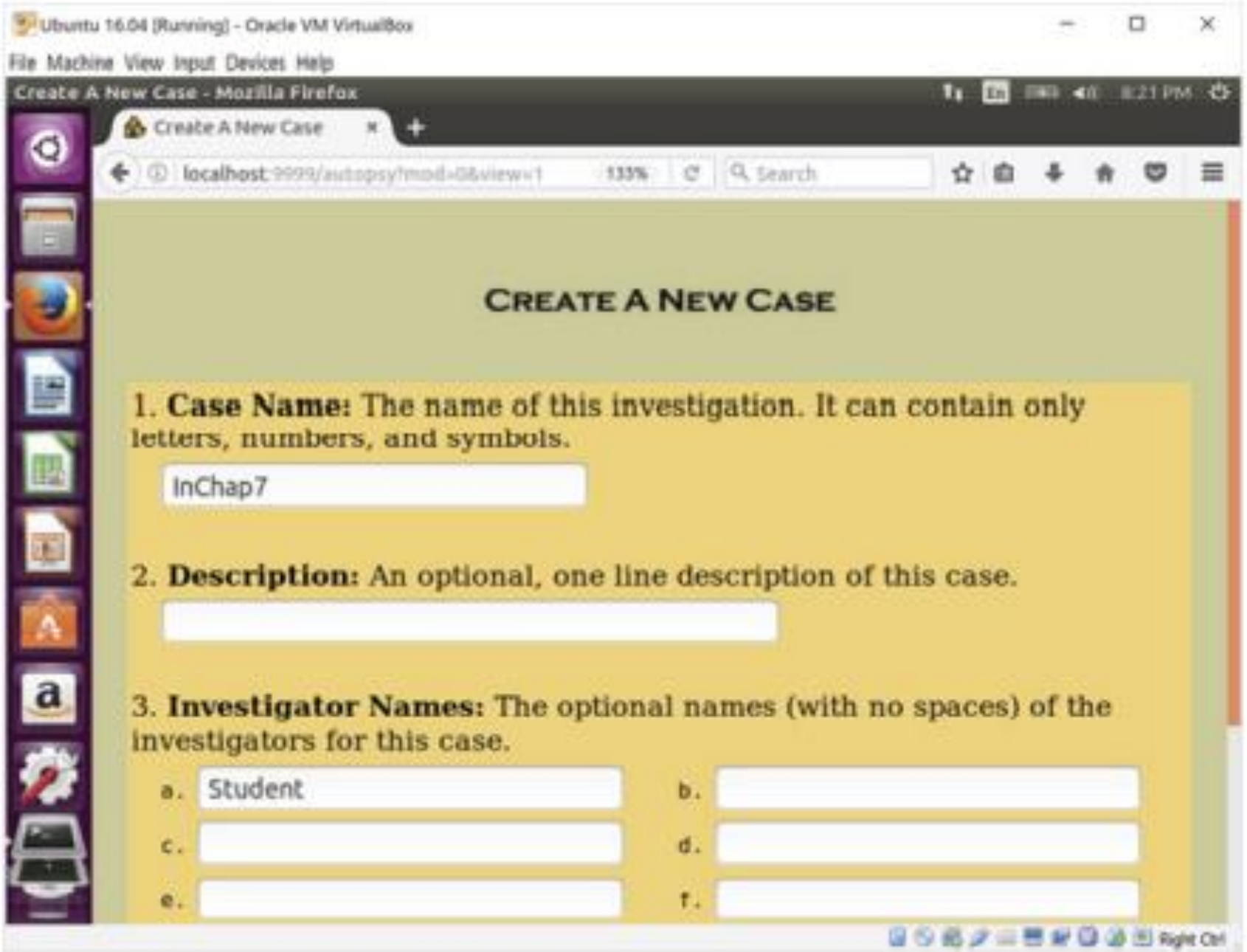

Autopsy Forensic Browser - Mozilla Firefox
WARNING: Your browser currently has Java Script enabled.
You do not need Java Script to use Autopsy and it is recommended that it be turned off for security reasons.
Autopsy Forensic Browser 2.24
Open Case
New Case
Help

- If you see a warning message at the top stating that JavaScript is enabled, you have to reconfigure your browser to disable it.
- After reconfiguring the browser, you might have to exit and restart.
- If the Autopsy terminal session is still running, simply paste the Autopsy URL into the Address text box again

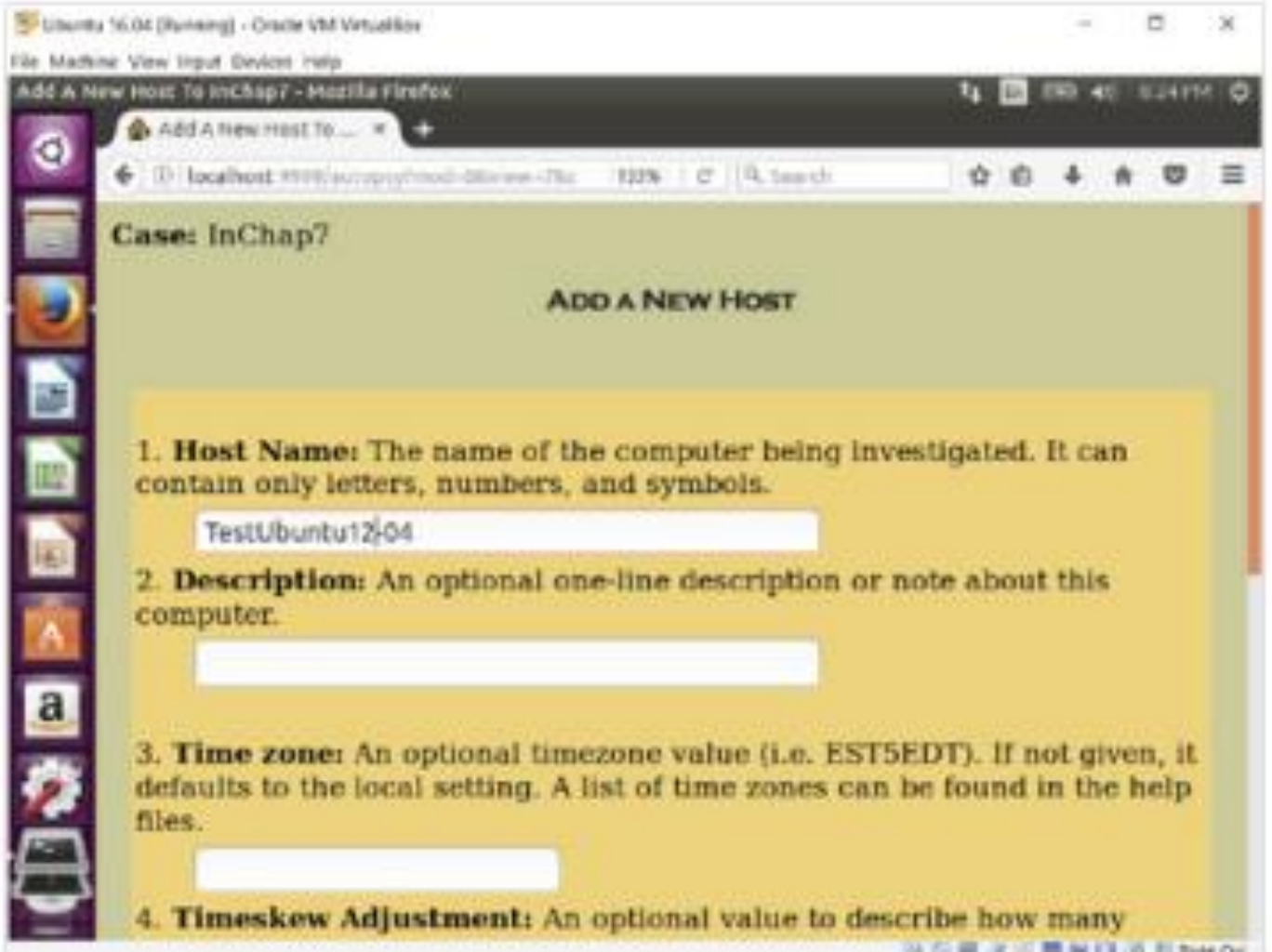
Examining a Case with Sleuth Kit and Autopsy

- In digital forensics, sometimes you have to reevaluate cases that are several years old, and this activity gives you a chance to do that.
- Before starting the examination, copy the GCFILX.00n (with n representing a number from 1 to 5) image files from your work folder to the evidence locker you set up in the previous activity.
- Autopsy uses the evidence locker to save results from examinations.

- To start the examination, follow these steps:
  - In Autopsy’s main window, click the New Case button shown previously in Figure
  - When the Create a New Case dialog box opens, enter InChap7 for the case name (see Figure), a description (optional), and your name, and then click the New Case button to continue.


Ubuntu 16.04 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Create A New Case - Mozilla Firefox
Create A New Case
localhost:9999/autopsy?mod=0&view=1
CREATE A NEW CASE
1. Case Name: The name of this investigation. It can contain only letters, numbers, and symbols.
InChap7
2. Description: An optional, one line description of this case.
3. Investigator Names: The optional names (with no spaces) of the investigators for this case.
a. Student
b.
c.
d.
e.
f.

- In the Creating Case dialog box, click Add Host to continue.
- In the Add a New Host dialog box, enter TestUbuntu12-04 for the hostname (see Figure 7-15), and then click Add Host.



- In the Adding host dialog box, click Add Image to continue. In the Open Image dialog box, click Add Image File.
- If you don’t click Partition in Step 4, the image is read as raw data, and file and directory structures aren’t visible to Autopsy.
- In the Add a New Image dialog box, type the complete path to the evidence locker in the Location text box, click the Partition and Move option buttons, and then click Next.
  - (Remember that Linux commands are case sensitive. If you enter a lowercase filename and the filename is uppercase, Autopsy can’t find and load the file.)

- In the Split Image Confirmation dialog box, verify that all images are correctly loaded; if they are, click Next. If not, click Cancel. (If this data is incorrect, it’s probably caused by an error in the pathname to the evidence locker or image files.)

- If you have multiple segment volumes that are sequentially numbered or lettered (the dd command with the split option without the -d switch), use an asterisk as the extension (for example, GCFI-LX.*) so that all segments are read sequentially.
- In the Image File Details section, click the Calculate the hash value for this image option button, and then click Add.
- In the Calculating MD5 message box, click OK. In the “Select a volume to analyze or add a new image file” dialog box, click Analyze and then Keyword Search to start a search for keywords of interest to the investigation.
- In the Keyword Search of Allocated and Unallocated Space dialog box, type the name martha in the text box, and then click Search.
- When the search is finished, Autopsy displays a summary of the search results. To see detailed search results, click the link to results link at the upper left

- Examine the search results by scrolling through the left pane, and then click the Fragment 236019 “Ascii” link to view details of the search.
- Repeat this examination by clicking other ASCII and Hex links for the remaining hits.
- When you’re finished examining the search hits, close the Searching for ASCII and Searching for Unicode dialog box to return to the “Select a volume to analyze or add a new image file” dialog box.
- Exit Autopsy, and log off Ubuntu.

Importance of Write-Blocker

- The first item you should consider for a forensic workstation is a write-blocker .
- Writeblockers protect evidence disks by preventing data from being written to them.
- Software and hardware write-blockers perform the same function but in a different fashion.

- Software write-blockers, such as PDBlock from Digital Intelligence, typically run in a shell mode
- PDBlock changes interrupt-13 of a workstation's BIOS to prevent writing to the specified drive.
- If you attempt to write data to the blocked drive, an alarm sounds, advising that no writes have occurred.
- PDBlock can run only in a true DOS mode, however, not in a Windows CLI.

- With hardware write-blockers, you can connect the evidence drive to your workstation and start the OS as usual.
- Hardware write-blockers, which act as a bridge between the suspect drive and the forensic workstation, are ideal for GUI forensics tools.
- They prevent Windows or Linux from writing data to the blocked drive.

- Many vendors have developed write-blocking devices that connect to a computer through FireWire, USB 2.0 and 3.0, SATA, PATA, and SCSI controllers.
- Most of these write-blockers enable you to remove and reconnect drives without having to shut down your workstation, which saves time in processing the evidence drive

Acquiring Data with a Linux Boot CD

- The Linux OS has many features that are applicable to digital forensics, especially data acquisitions.
- In Windows OSs and newer Linux kernels, when you connect a drive via USB, FireWire, external SATA, or even internal PATA or SATA controllers, both OSs automatically mount and access the drive.

Using Linux Live CD Distributions

- Several Linux distributions, such as Ubuntu, openSUSE, Arch Linux, Fedora, and Slackware, provide ISO images that can be burned to a CD or DVD.
- They’re called “Linux Live CDs.”
- Most of these Linux distributions are for Linux OS recovery, not for digital forensics acquisition and analysis.

- A few Linux ISO images are designed specifically for digital forensics, however.
- These images contain additional utilities that aren’t typically installed in normal Linux distributions.
- They’re also configured not to mount, or to mount as read-only, any connected storage media, such as USB drives.

- This feature protects the media’s integrity for the purpose of acquiring and analyzing data.
- To access media, you have to give specific instructions to the Live CD boot session through a GUI utility or a shell command prompt.
- Mounting drives from a shell gives you more control over them.

- Linux can read data from a physical device without having to mount it.
- As a usual practice, don’t mount a suspect media device as a precaution against any writes to it

The following are some well-designed Linux Live CDs for digital forensics:

- Penguin Sleuth Kit
- CAINE
- Deft
- Kali Linux previously known as BackTrack
- Knoppix
- SANS Investigate Forensic Toolkit

- After creating a Linux Live CD, test it on your workstation.
- Remember to check your workstation’s BIOS to see whether it boots first from the CD or DVD on the system.
- To test the Live CD, simply place it in the CD or DVD drive and reboot your system.
- If successful, Linux loads into your computer’s memory, and a common GUI for Linux is displayed.

How to use Linux to make forensically sound data acquisitions.

Preparing a Target Drive for Acquisition in Linux

- The Linux OS has many tools you can use to modify non-Linux file systems.
- Current Linux distributions can create Microsoft FAT and NTFS partition tables.

Acquiring Data with dd in Linux

- A unique feature of a forensics Linux Live CD is that it can mount and read most drives. To perform a data acquisition on a suspect computer, all you need are the following:
  - A forensics Linux Live CD
  - A USB, FireWire, or SATA external drive with cables
  - Knowledge of how to alter the suspect computer's BIOS to boot from the Linux Live CD
  - Knowledge of which shell commands to use for the data acquisition

- The dd command, available on all UNIX and Linux distributions, means “data dump.”
- This command, which has many functions and switches, can be used to read and write data from a media device and a data file.
- The dd command isn’t bound by a logical file system’s data structures, meaning the drive doesn’t have to be mounted for dd to access it.

- For example, if you list a physical device name, the dd command copies the entire device—all data files, slack space, and free space (unallocated data) on the device.
- The dd command creates a raw format file that most forensics analysis tools can read, which makes it useful for data acquisitions.

- As powerful as this command is, it does have some shortcomings.
- One major problem is that it requires more advanced skills than the average computer user might have.
- Also, because it **doesn’t compress data**, the target drive needs to be equal to or larger than the suspect drive.
- It’s possible to divide the output to other drives if a large enough target drive isn’t available, but this process can be cumbersome and prone to mistakes when you’re trying to keep track of which data blocks to copy to which target drive.

- The **dd command combined with the split command** segments output into separate volumes.
- Use the split command with the -b switch to adjust the size of segmented volumes the dd command creates.
- As a standard practice for archiving purposes, create segmented volumes that fit on a CD or DVD.

- Follow these steps to make an image of an NTFS disk on a FAT32 disk by using the dd command:
- 1. Assuming that your workstation is the suspect computer and is booted from a Linux Live CD, connect the USB, FireWire, or SATA external drive containing the FAT32 target drive, and turn the external drive on.
- 2. If you’re not at a shell prompt, start a shell window, switch to superuser (su) mode, type the root password, and press Enter.

- 3. At the shell prompt, list all drives connected to the computer by typing fdisk -l and pressing Enter, which produces the following output:

```
Disk /dev/hda: 40.0 GB, 40007761920 bytes
255 heads, 63 sectors/track, 4864 cylinders
Units - cylinders of 16065 * 512 - 8225280 bytes

Device Boot    Start     End      Blocks    Id   System
/dev/hda1   *     1        13      104391    83   Linux
/dev/hda2        14      4864    38965657+   8e   Linux LVM

Disk /dev/sda: 163.9 GB, 163928605184 bytes
255 heads, 63 sectors/track, 19929 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

Device Boot    Start     End      Blocks    Id  System
/dev/sda1        1       12000    96389968+   b  W95 FAT32
/dev/sda2    12001       19929    63689692+   5  Extended
/dev/sda5    12001       19929    63689661    c  W95 FAT32 (LBA)

Disk /dev/sdb: 6448 MB, 6448619520 bytes
199 heads, 62 sectors/track, 1020 cylinders
Units = cylinders of 12338 * 512 = 6317056 bytes

Device Boot    Start    End    Blocks   Id   System
/dev/sdb1        1      1020   6292349    7   HPFS/NTFS
```

- 4. To create a mount point for the USB, FireWire, or SATA external drive and partition, make a directory in /mnt by typing mkdir /mnt/sda5 and pressing Enter.
- 5. To mount the target drive partition, type mount -t vfat /dev/sda5 /mnt/sda5 and press Enter.
- 6. To change your default directory to the target drive, type cd /mnt/sda5 and press Enter.

- 7. List the contents of the target drive’s root level by typing ls -al and pressing Enter. Your output should be similar to the following:

```
total 40
drwxr-xr-x  2 root root 32768 Dec 31  1969 .
drwxr-xr-x  5 root root  4096 Feb  6 17:22 ..
```

- 8.To make a target directory to receive image saves of the suspect drive, type mkdir case01 and press Enter.
- 9. To change to the newly created target directory, type cd case01 and press Enter. Don’t close the shell window.

- Next, you perform a raw format image of the entire suspect drive to the target directory.
- To do this, you use the split command with the dd command.
- The split command creates a two-letter extension for each segmented volume. The -d switch creates numeric rather than letter extensions.

- As a general rule, if you plan to use a Windows forensics tool to examine a dd image file created with this switch, the segmented volumes shouldn’t exceed 2 GB each because of FAT32 file size limits. This 2 GB limit allows you to copy only up to 198 GB of a suspect’s disk.
- If you need to use the dd command, it’s better to use the split command’s default of incremented letter extensions and make smaller segments.

- 1. Type dd if=/dev/sdb | split -b 650m - image_sdb. and press Enter. You should see output similar to the following:

```
12594960+0 records in
12594960+0 records out
```

- 2. List the raw images that have been created from the dd and split commands by typing ls -l and pressing Enter. You should see output similar to the following:

```
total 6297504
-rwxr-xr-x  1 root root 681574400 Feb  6 17:26 image_sdb.aa
-rwxr-xr-x  1 root root 681574400 Feb  6 17:28 image_sdb.ab
-rwxr-xr-x  1 root root 681574400 Feb  6 17:29 image_sdb.ac
-rwxr-xr-x  1 root root 681574400 Feb  6 17:30 image_sdb.ad
-rwxr-xr-x  1 root root 681574400 Feb  6 17:32 image_sdb.ae
-rwxr-xr-x  1 root root 681574400 Feb  6 17:33 image_sdb.af
-rwxr-xr-x  1 root root 681574400 Feb  6 17:34 image_sdb.ag
-rwxr-xr-x  1 root root 681574400 Feb  6 17:36 image_sdb.ah
-rwxr-xr-x  1 root root 681574400 Feb  6 17:37 image_sdb.ai
-rwxr-xr-x  1 root root 314449920 Feb  6 17:37 image_sdb.aj
```

- 3. To complete this acquisition, dismount the target drive by typing umount /dev/sda5 and pressing Enter.

- Depending on the Windows forensics analysis tool you're using, renaming each segmented volume's extension with incremented numbers instead of letters might be necessary.
- For example, rename image_sdb.aa as image_sdb.01, and so on. Several Windows forensics tools can read only disk-to-image segmented files that have numeric extensions.
- Most Linux forensics tools can read segments with numeric or lettered extensions.

Acquiring Data with dcfldd in Linux

- The dd command is intended as a data management tool; it's not designed for forensics acquisitions.
- Because of these shortcomings, Nicholas Harbour of the Defense Computer Forensics Laboratory (DCFL) developed a tool that can be added to most UNIX/Linux OSs.
- This tool, the dcfldd command, works similarly to the dd command but has many features designed for forensics acquisitions.

The following are important functions dcfldd offers that aren’t possible with dd:

- Specify hexadecimal patterns or text for clearing disk space.
- Log errors to an output file for analysis and review.
- Use the hashing options MD5, SHA-1, SHA-256, SHA-384, and SHA-512 with logging and the option of specifying the number of bytes to hash, such as specific blocks or sectors.

- Refer to a status display indicating the acquisition’s progress in bytes.
- Split data acquisitions into segmented volumes with numeric extensions.
- Verify the acquired data with the original disk or media data.

- When using dcfldd, you should follow the same precautions as with dd.
- The dcfldd command can also write to the wrong device, if you aren’t careful.

- The following examples show how to use the dcfldd command to acquire data from a 64 MB USB drive, although you can use the command on a larger media device.
- All commands need to be run from a privileged root shell session.
- To acquire an entire media device in one image file, type the following command at the shell prompt:

```
dcfldd if=/dev/sda of=usbimg.dat
```

- If the suspect media or disk needs to be segmented, use the dcfldd command with the split command, placing split before the output file field (of=), as shown here:

```
dcfldd if=/dev/sda hash=md5 md5log=usbimgmd5.txt bs=512
  conv=noerror,sync split=2M of=usbimg
```

- This command creates segmented volumes of 2 MB each. To create segmented volumes that fit on a CD of 650 MB, change the split=2M to split=650M.
- This command also saves the MD5 value of the acquired data in a text file named usbimgmd5.txt.

Validating Data Acquisitions

- Probably the most critical aspect of computer forensics is validating digital evidence.
- The weakest point of any digital investigation is the integrity of the data you collect, so validation is essential.

- Validating digital evidence requires using a hashing algorithm utility, which is designed to create a binary or hexadecimal number that represents the uniqueness of a data set, such as a file or disk drive.
- This unique number is referred to as a “digital fingerprint.”
- With a few exceptions, making any alteration in one of the files—even changing one letter from uppercase to lowercase—produces a completely different hash value.

- These exceptions, known as “collisions,” have been found to occur in a small number of files with MD5, and SHA-1 might also be subject to collisions.
- For forensic examinations of data files on a disk drive, however, collisions are of little concern.
- If two files with different content have the same MD5 hash value, a comparison of each byte of a file can be done to see the differences.

- Currently, several tools can do a byte-by-byte comparison of files.
- Programs such as X-Ways Forensics, X-Ways WinHex, and IDM Computing Solution’s UltraCompare can analyze and compare data files.

- For imaging an evidence drive, many tools offer validation techniques ranging from CRC-32, MD5, and SHA-1 to SHA-512.
- These hashing algorithm utilities are available as stand-alone programs or are integrated into many acquisition tools.

Linux Validation Methods

- Linux is rich in commands and functions.
- The two Linux shell commands dd and dcfldd, have several options that can be combined with other commands to validate data.
- The dcfldd command has other options that validate data collected from an acquisition.
- Validating acquired data with the dd command requires using other shell commands.

- Current distributions of Linux include two hashing algorithm utilities: md5sum and sha1sum.
- Both utilities can compute hashes of a single file, multiple files, individual or multiple disk partitions, or an entire disk drive

Validating dd-Acquired Data

- As shown earlier, the following command produces segmented volumes of the /dev/sdb drive, with each segmented volume named image_sdb and an incrementing extension of .aa, .ab, .ac, and so on:

```
dd if=/dev/sdb | split -b 650m image_sdb
```

- To validate all segmented volumes of a suspect drive with the md5sum utility, you use the Linux shell commands shown in the following steps.
- For the saved images, remember to change to the directory where the data was saved, or list the exact path for the saved images.
- To use sha1sum instead of md5sum, just replace all md5sum references in commands with sha1sum.
- The drive should still be connected to your acquisition workstation.

1. If necessary, start Linux, open a shell window, and navigate to the directory where image files are saved. To calculate the hash value of the original drive, type md5sum/dev/sdb > md5_sdb.txt and press Enter.

2. To compute the MD5 hash value for the segmented volumes and append the output to the md5_sdb.txt file, type cat image_sdb.* | md5sum >> md5_sdb.txt and press Enter.

3. Examine the md5_sdb.txt file to see whether both hashes match by typing cat md5_sdb.txt and pressing Enter. If the data acquisition is successful, the two hash numbers should be identical. If not, the acquisition didn’t work correctly. You should see output similar to the following:

```
34963884a4bc5810b130018b00da9de1   /dev/sdb
34963884a4bc5810b130018b00da9de1
```

4. Close the Linux shell window by typing exit and pressing Enter.

- With the dd command, the md5sum or sha1sum utilities should be run on all suspect disks and volumes or segmented volumes.

Validating dcfldd-Acquired Data

- Because dcfldd is designed for forensics data acquisition, it has validation options integrated: hash and hashlog.
- You use the hash option to designate a hashing algorithm of md5, sha1, sha256, sha384, or sha512.

- The hashlog option outputs hash results to a text file that can be stored with image files.
- To create an MD5 hash output file during a dcfldd acquisition, you enter the following command (in one line) at the shell prompt:

```
dcfldd if=/dev/sda split=2M of=usbimg hash=md5
  hashlog=usbhash.log
```

- To see the results of files generated with the split command, you enter the list directory (ls) command at the shell prompt.
- You should see the following output:

```
usbhash.logusbimg.004 usbimg.010 usbimg.016 usbimg.022 usbimg.028
usbseghash.logusbimg.005 usbimg.011 usbimg.017 usbimg.023 usbimg.029
usbimg.000 usbimg.006 usbimg.012 usbimg.018 usbimg.024 usbimg.030
usbimg.001 usbimg.007 usbimg.013 usbimg.019 usbimg.025
usbimg.002 usbimg.008 usbimg.014 usbimg.020 usbimg.026
usbimg.003 usbimg.009 usbimg.015 usbimg.021 usbimg.027
```

- Another useful dcfldd option is vf (verify file), which compares the image file with the original medium, such as a partition or drive.
- The vf option applies only to a nonsegmented image file.
- To validate segmented files from dcfldd, use the md5sum or sha1sum command described previously.
- To use the vf option, you enter the following command at the shell prompt:

```
dcfldd if=/dev/sda vf=sda_hash.img
```

Windows Validation Methods

- Unlike Linux, Windows has no built-in hashing algorithm tools for digital forensics.
- However, many Windows third-party programs do have a variety of built-in tools.
- These third-party programs range from hexadecimal editors, such as X-Ways WinHex or Breakpoint Software Hex Workshop, to forensics programs, such as OSForensics, Autopsy, EnCase, and FTK.

- Commercial forensics programs also have built-in validation features.
- Each program has its own validation technique used with acquisition data in its proprietary format.
- For example, Autopsy uses MD5 to validate an image.
- It reads the metadata in Expert Witness Compression or AFF image files to get the original hash.
- If the hashes don’t match, Autopsy notifies you that the acquisition is corrupt and can’t be considered reliable evidence.

- In Autopsy and many other forensics tools, however, raw format image files don’t contain metadata.
- As mentioned, a separate manual validation is recommended for all raw acquisitions at the time of analysis.
- The previously generated validation file for raw format acquisitions is essential to the integrity of digital evidence.
- The saved validation file can be used later to check whether the acquisition file is still good.