# Chapter 3
# Building a Forensics Workstation


**Learning Objectives**
The objectives of this chapter are to:

- Build a computer forensics workstation using open source tools
- Use TSK and Autopsy to conduct a digital forensics investigation

If you've seen CSI: NY (Crime Scene Investigation: New York) or any other detective TV shows, you will notice that detectives use tools to search and find clues and evidence within the crime scene. For example, they use tweezer to pick up small objects on the ground and camera to take pictures of crime scenes. As the age of technology, many crimes involve using technological devices. This is why the law enforcement needs to embrace digital forensics. Besides traditional forensic equipment and tools, detectives will now need digital forensics tools to analyze electronic devices and media. In this chapter, we will build a computer forensics workstation using open source forensic tools. Particularly, we have chosen The Sleuth Kit (TSK) and Autopsy for our book, as it is a widely used open source forensic toolkit. Also, we will learn how to use TSK and Autopsy Forensics Browser to conduct a digital forensics investigation through practical exercises.

## 3.1 The Sleuth Kit (TSK) and Autopsy Forensic Browser

### *3.1.1 The Sleuth Kit (TSK)*

The Sleuth Kit, better known as TSK is a collection of free forensic tools. It was developed by Brian Carrier [1] and is available at http://www.sleuthkit.org/. This is a forensic suite available for Linux distribution and is used primarily to analyze file



X. Lin, *Introductory Computer Forensics*,
https://doi.org/10.1007/978-3-030-00581-8_3

systems and storage media. TSK can be installed via package manage or compiled from source code. Forensic Linux distributions, like BackTrack, Kali, Helix or Penguin Sleuth Kit, ship with TSK pre-installed. If you are operating from a Windows workstation you can use TSK via a virtual machine installation. TSK tools are used on command line interfaces. Autopsy, developed by Basis technology, is a graphical front-end for TSK. The aim of TSK is to create the leading forensic analysis tool for source file and volume systems available on all major platforms.

Brian Carrier developed TSK in collaboration with @stake. It was initially known as @stake The Sleuth Kit or TASK. TASK was created to fill many gaps found on two popular tools for digital forensic analysis, The Coroner's Toolkit (TCT) [2] and TCTUTILS [3]. TASK added support for FAT and NTFS file systems. The predecessors of TASK and TSK, TCT and TCTUTILS were first developed in 2000 by Dan Farmer and Wieste Venema. TCT was an innovative approach to digital forensics. It was free and open sourced and was available to the public. However, TCT file system tools could only support operations on the inode or block layer. During analysis, file directory names were not utilized. Furthermore, TCT was platform dependent. In other words, analysis could only be performed on a filesystem if it was the same version as that performing the analysis. This caveat made it difficult to create forensics OS distributions that we have today.

TSK provides a large number of specialized command-line based utilities. It is capable of parsing many types of file systems, including Ext2, Ext3, Ext4, HFS, ISO 9660, UFS ½, YAFFS2, FAT/ExFAT, and NTFS file systems. It can analyze within disk images stored in raw images that are in dd format, AFF formats or Expert Witness formats. You can use TSK using command line tools or as a library embedded within a separate digital forensic tool such as Autopsy. TSK was originally designed to tackle digital forensics with a layered approach. The tools in the original TASK distribution were developed to tackle specific layers of a forensic image. We can separate forensic images into four main and distinct layers [4].

1. File System Layer
2. Content /Data Layer
3. Metadata/Inode Layer
4. File Layer

The File System Layer consists of disks used in digital forensics. These disks consist of one or more slices; otherwise known as partitions. These partitions contain their own file systems. There are several types of file systems. Popular are File Allocation Table (FAT), fourth extended filesystem (ext4) and New Technologies File System (NTFS). Values that allow you to differentiate among other file systems are contained on this layer. TSK tools for this layer are prefixed with 'fs'. File System tools are used to display general file system details. This includes layouts, allocation structures and boot blocks.

Data is stored in pieces. These pieces can be called blocks, fragments or clusters depending on how data is stored. The Content or Data Layer houses file and directory content. Tools for this layer are prefixed by 'blk'. Previous versions of TSK and TASK used the prefix 'd'. These tools are geared towards the search and recovery of actual information and can be crucial in the recovery of deleted content.

The Metadata or Inode Layer stores descriptive information. This includes inode structures or entries for various file systems or platforms. These include directory and MFT entries from FAT and NTFS respectively and inodes from Ext and UFS. Furthermore, timestamp, addresses and size data can be collected on this level. TSK tools for this level are prefixed with an 'i'.

The final layer is known as the File Layer. It sometimes referred to as the Human Interface Level. This level allows for interaction between users and file content. File names are saved in data units which are allocated by parent directories. File Name structures contain the name and addresses of to a corresponding metadata structure. TSK tools for the File Layer are prefixed with 'f'. File Name Layer handles name structures. This is useful in gathering data based on the name of files. However, file names and directory structures do not often fully demonstrate the content of files. File Name Layer tools are useful in cataloging the contents of a volume.

TSK hosts several tools that fall outside or work between layers. These can be categorized as Fully Automated, File System Journal, Volume System, Image File, Disk and other miscellaneous tools. Description of all tools are given in the table below.

Tools in the TSK Suite [5] are listed below.

| Tool category | Tool name | Description |
|---|---|---|
| File system layer | fsstat | This command is used to display all details associated with a file system |
| File name layer | ffind | This command is used to find unallocated and allocated file names that point to specific meta data structure |
| | fls | Lists names in a directory. These include deleted file names as well |
| Meta data layer | icat | Used to extract data units from a file as per meta data address rather than the file name |
| | ifind | Used to find the meta data structure that has a given file name or other meta data structure pointing to it |
| | ils | Used to list meta structures and their content |
| | istat | Used to display statistics. Specifically, statistics on meta data structures |
| Data unit layer | blkcat | Extract and display the contents of a given data unit |
| | blkls | Used to list details concerning data units. Can also detail which data units are allocated or not |
| | blkstats | Used to display statistics on given data structures |
| | blkcalc | Used calculate where data found in unallocated space can be found on the original image |
| File system journal layer | jcat | Display information of a journal block |
| | jls | List entries for a file system journal |
| Volume system layer | mmls | This command is used to display disk layout and organization |
| | mmstat | Used to display information on the volume system |
| | mmcat | Used to extract contents from a partition |

(continued)

| Tool category | Tool name | Description |
|---|---|---|
| Image file layer | img_stat | Displays the details of an image file. Used to collect size of images and the byte range of split image formats |
| | img_cat | Used to output the contents of image files. Displays the raw content of image files |
| Disk tools layer | disk_sreset | This tool is used to remove Host Protected Areas (HPA) if they exist |
| | disk_stat | Displays if HPAs exist on an image |
| Automated tools | tsk_comparedir | Used to compare local directories with images or raw devices. This can be used to detect if rootkits are used to hide files from the local directory hierarchy. TSK parses raw content from the raw device |
| | tsk_gettimes | Extracts metadata to be used by mactime to create timelines. This is useful for timeline analysis |
| | tsk_loaddb | Saves the volume, image, and file metadata to a SQLite database. This database can then be used by other non-TSK programs for further analysis |
| | tsk_recover | Used to extract unallocated and allocated files from an image. The files can then be saved to a local directory |
| Miscellaneous | hfind | Uses a binary sort algorithm and compares them to hashes found in hash databases. Hashes are md5sum |
| | mactime | Creates a timeline for a file's activity |
| | sorter | Sorts files based on file type. Also, performs extension checking and hash database lookups. Useful in checking whether file extensions have been changed to secret contents |
| | sigfind | This command is used to find binary signatures in a given data set |

### *3.1.2 Autopsy Forensic Browser*

Autopsy forensic browser or simply Autopsy, also known as Autopsy server or Autopsy service, is a digital forensics platform and graphical interface to TSK and other digital forensics tools usually used by the military or corporate examiners for investigation on a victim's computer. As Autopsy is HTML-based, you can connect to it from any platform using an HTML browser, for example, Firefox. Autopsy provides a "File Manager"-like interface, which gives investigators a convenient way to managing their investigation cases, showing the details about deleted data and file system structures of imported disk (or partition) images [6]. Simply to say, Autopsy forensic browser is easy to set up in a Linux system as the only way to accessing this browser is the input of the URL http://localhost:9999/autopsy. For example, you should be able to launch the Autopsy Forensic Browser in Kali Linux by navigating to Applications → Forensics → autopsy, shown in Fig. 3.1, and then connect to the Autopsy server by opening a Web browser and typing the above URL in the URL bar.

Afterwards, the default start page is displayed like Fig. 3.2, and you can start a digital forensics investigation by either creating a new case or opening an existing one.

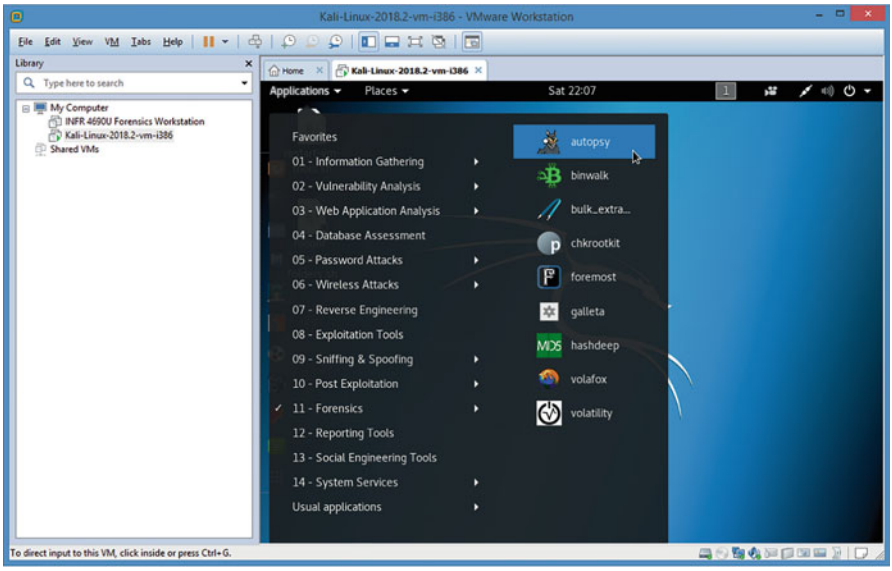



**Fig. 3.1** Launch Autopsy in Kali Linux


**Fig. 3.2** Autopsy web GUI

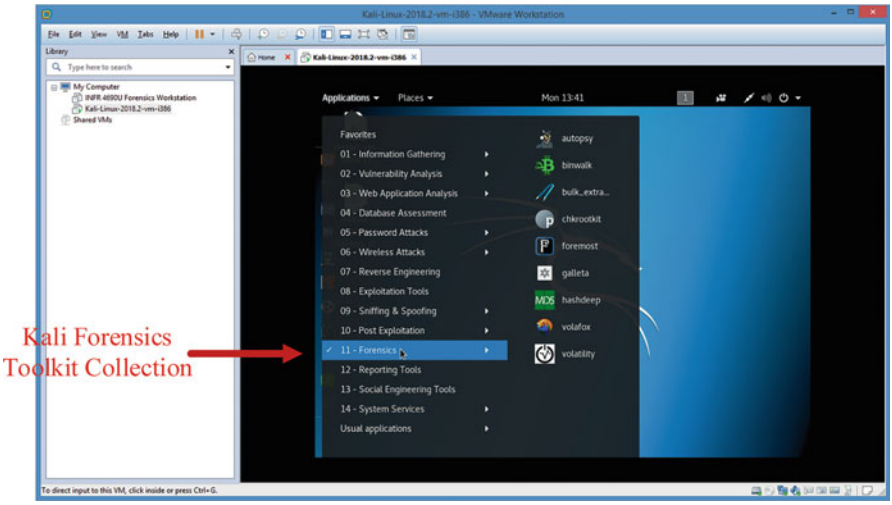



**Fig. 3.3** Kali Linux

### *3.1.3 Kali Linux Sleuth Kit and Autopsy*

Kali Linux, with its BackTrack lineage, is a digital forensics and penetration testing Linux distribution. It is based on Debian Linux, and has over 600 preinstalled digital forensics and penetration-testing programs, including TSK and Autopsy (Fig. 3.3). We will use Kali Linux to build a Forensics Workstation for our book. There are still many other interesting tools available online, such as The SANS Investigative Forensic Toolkit (SIFT) [7]. The SANS SIFT kit is a computer forensics VMware appliance pre-configured with all the necessary tools for digital forensic examinations.

## 3.2 Virtualization

The virtualization technology has been introduced as a solution that several operating systems and applications can be run on one physical computer, known as "host", in order to address one limitation of today's computers, which are designed to run just one operating system at a time. Each self-contained "virtual machine" runs like a separate physical computer, and has its own virtualized computing resources, including virtual CPU, virtual hard disks, virtual memory, based on its requirements to computing resources available on the host. Whereas OS installed and running on a physical server is referred to as primary operating system, each VM runs its own operating system, called guest operating system.

Simply put, virtualization is an abstraction layer in the computer architecture model where an operating system communicates with this layer, instead of directly

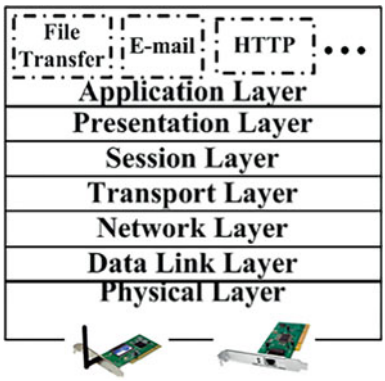



**Fig. 3.4** ISO Open Systems Interconnection (OSI) model

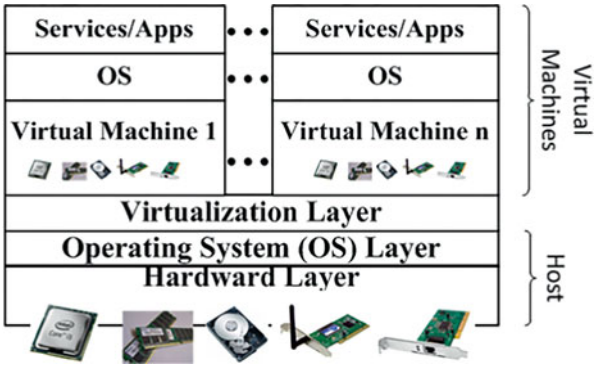



**Fig. 3.5** The virtualization layer sitting over the host OS and letting you run multiple virtual systems each with its own OS and services/applications

communicating with the hardware. An abstraction layer can be described as a way of dividing up and isolating a model based on functionality. An example of such a concept would be the Open Systems Interconnection (OSI) Model, shown in Fig. 3.4, where the seven layers are split up based on their functions.

In a usual case, a person would run an operating system in a "virtual machine", shown in Fig. 3.5, which would basically create an environment which emulates an actual computer, allowing the operating system to function normally within the limits set in the virtual machine. Nevertheless, besides operating system, technically, virtualization could be the "creation of a virtual (rather than actual) version of any computing system, including a server, a storage device or network resources" [8].

### *3.2.1 Why Virtualize?*

The benefits of using virtualization (e.g., a pre-configured Fodera virtual machine) include

First, we can save a lot of time from configuring the devices and softwares. If thing doesn't work out, we can always roll back to a snapshot and start over again until it works. In other words, we can have an environment that can be saved, deleted, backed up, etc., on demand. By using virtualization, we can always have a copy of clean and workable image, which is very good for the purpose of teaching.

Second, all students have the same lab environments, which can be well controlled. As a result, it could become easy to troubleshoot and diagnose problems in the lab environments of students.

Third, another reason to virtualize is to have a testing environment that can be saved, deleted, backed up, etc., on demand.

Finally, as for virtual machines, thin provisioning is used for just enough physical space being used as needed, and allows you to create virtual hard disks of a certain size without occupying as much space (it consumes space as it needs it), allowing you to overcompensate the size of hard drives.

### *3.2.2 What Are the Virtualization Options?*

The virtualizing platform (e.g. VMware Workstation, Oracle VirtualBox, Citrix XenServer, etc.) creates an almost-transparent layer between the hardware and the operating systems running. The platform also creates virtual hard disks, virtual CPU's, etc. needed for the virtual operating system to run. There are two common types of virtualizing platforms:

1. Ones that Run On an operating system
   *For example*: Oracle VirtualBox runs as an application under Ubuntu.
   *Products*: VMware Workstation/Fusion, Oracle VirtualBox, Parallel's Desktop/Workstation (Fig. 3.6).
2. Ones that Run as an Operating System as a Hypervisor.
   *For example*: VMware ESXi runs as its own operating system, which is a very thin version of Linux, customized for virtualization, and the virtual operating systems communicate with the hypervisor.
   *Products*: VMware vSphere/ESXi, Citrix XenServer, Parallel's Virtuozzo Containers, Microsoft Hyper-V.

### *3.2.3 Why VMware Virtualization Platform?*

There are now many virtualization platforms available, including such as VMware, Microsoft Virtual PC, and Oracle VirtualBox. In the book, we choose VMware due

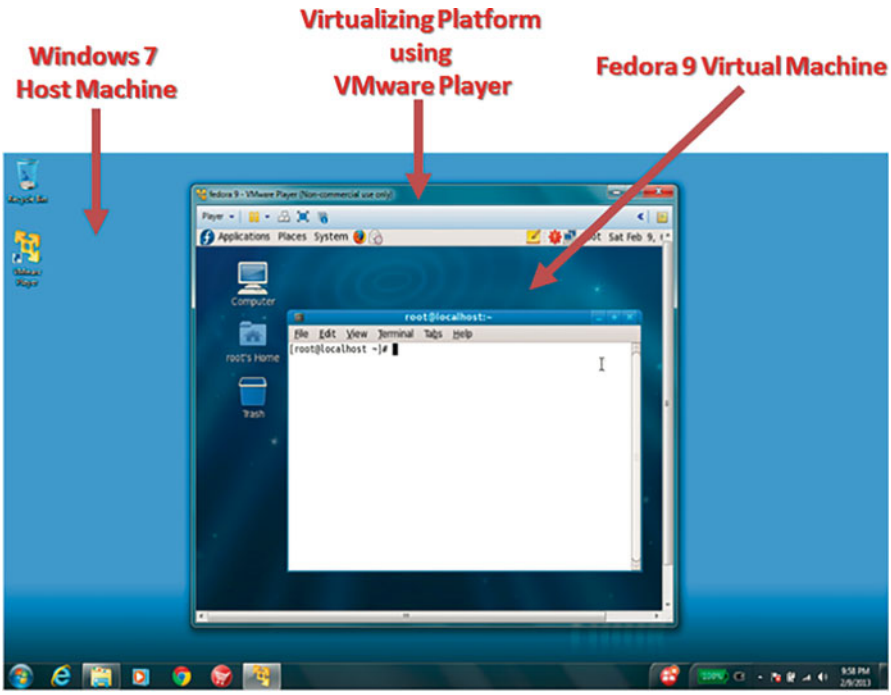



**Fig. 3.6** An example of virtualization environment where a Fedora 9 virtual machine running on Windows 7 machine using VMware Player (and configuring VMware Tools)

to the fact that VMware has a rich product line which allows us to deliver all the practice exercises developed in the book in a more flexible way to meet the needs of different institutions.

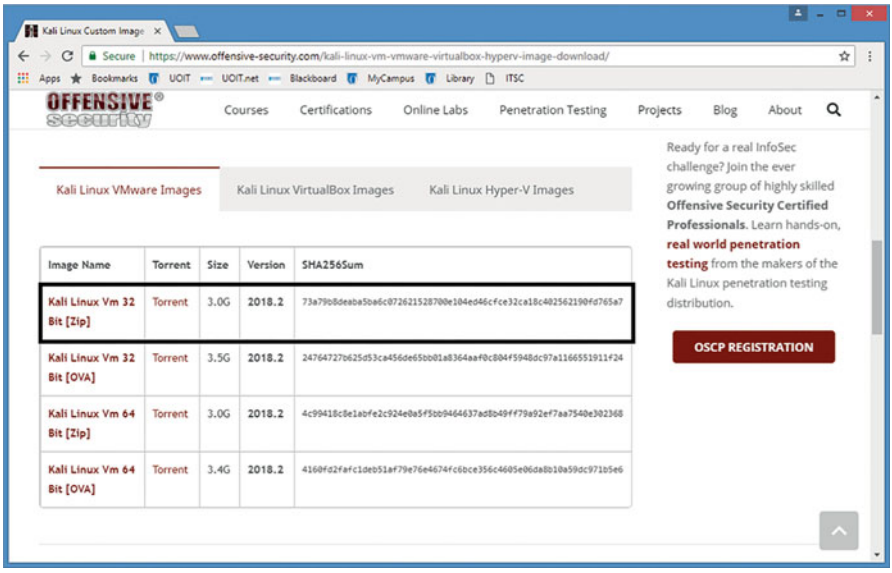
## 3.3 Building Up Your Forensics Workstation with Kali Linux

Next, you will build a Forensics Workstation using virtualization technologies and Kali Linux.

1. Download and install VMware Workstation Player (or VMware Workstation (PRO)) on your computer by going to http://www.vmware.com/

Note that VMware Workstation Player is free software that enables PC users to easily run any virtual machine on a Windows or Linux PC, but you may need to register with VMware using your email for the use of the software.

2. Download Kali Linux (Kali Linux 32 bit Vmware Preinstalled Image) by going to https://www.offensive-security.com/kali-linux-vmware-virtualbox-image-download/

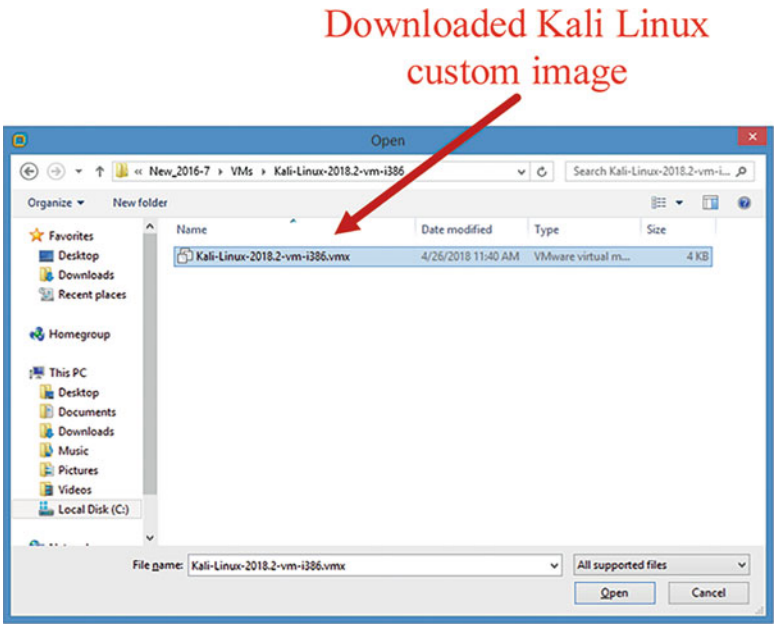



3. Install Kali Linux VMware Image in VMware

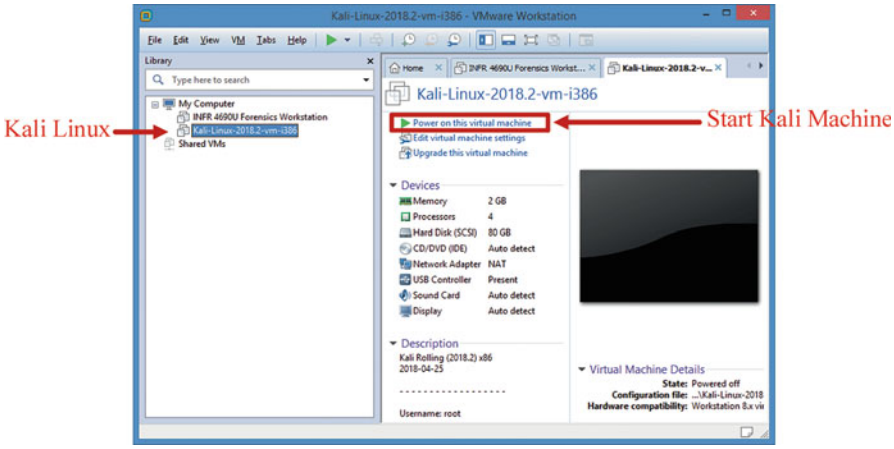
   Next, we will install Kali Linux VMware image in VMware Workstation/ VMware Player. The installation procedure is the same for VMware Workstation and VMware Player. Here we use VMware Workstation. Note that the downloaded VMware Image is a zipped file so we have to first extract the Virtual Machine files for Kali Linux VM.

   (a) Start VMware Workstation and then click on “File” and then click on “Open . . .”.

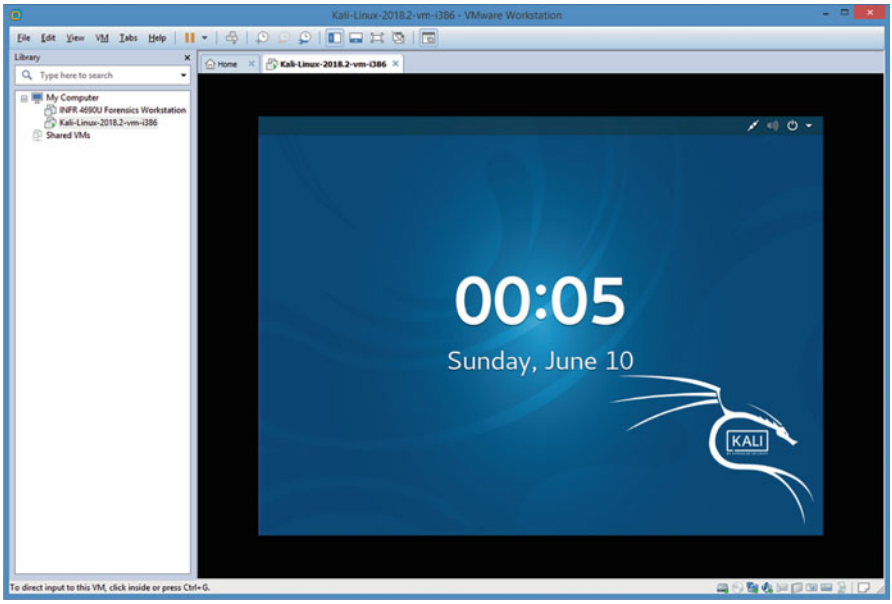
   (b) Now, browse to the folder for extracted Virtual Machine files and select the . vmx Kali Linux image file and click on “Open“.




(c) Click Open. Once the import gets completed then we will see the newly imported Kali Linux VM appeared in the list of available virtual machines on the left side of the VMware Workstation as like below:




(d) Click the Power on button to start Kali machine




Note that default root password for Kali Linux is "toor", without the quotes. Also, most of us like the convenience of using PuTTY for SSHing into a linux machine (herein Kali Linux VM). However, Kali Linux VM images have the SSH server disabled by default, though the SSH server is installed by default. To enable SSH server on Kali, log in to the Kali Linux VM as root from the console, and start a terminal and type the following commands

**First, Generate New Keys for Kali Linux SSH Server** For security reasons, it would better not use default keys provided in SSH server. Instead, you should back up the original keys (these files' names starting with "ssh_host_"), which can be found in the folder of /etc/ssh, and generate new keys for your Kali Linux SSH Server.

```
# cd /etc/ssh
# dpkg-reconfigure openssh-server
Creating SSH2 RSA key; this may take some time ...
2048 SHA256:ZYMY3yvTNuqjp27htTqyxsr5LQFW9l/4yO16/bdxVhc
root@kali (RSA)
Creating SSH2 ECDSA key; this may take some time ...
256 SHA256:JMhFkF26jMSZss6UTiPoF88gGBZ6vesmZi5fAyrXQAc
root@kali (ECDSA)
Creating SSH2 ED25519 key; this may take some time ...
256 SHA256:bRahoYuCNnjx+eeSc5VPM2T4ecpMYLnvRdTAnFycZJg
root@kali (ED25519)
```

Once it is done, the new keys have been generated.

**Second, Enable SSH Root Login** Edit the SSH server configuration file

```
# vi /etc/ssh/sshd_config
```

, change the following line to enable logging in through ssh as root
#PermitRootLogin prohibit-password
to
PermitRootLogin yes

**Finally, Start and Restart the Kali Linux SSH Server** start the Kali Linux SSH Server

```
# service ssh start
```

or, restart the Kali Linux SSH Server

```
# service ssh restart
```

Aslo, it is recommended to permanently enable the SSH service to start whenever Kali Linux VM starts, and type the following command

```
# systemctl enable ssh.service
Synchronizing state of ssh.service with SysV service script with/lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable ssh
Created symlink/etc/systemd/system/sshd.service → /lib/systemd/system/ssh.service.
```

4. Check the version of TSK installed on Kali Linux

```
# mmls -V
The Sleuth Kit ver 4.6.0
```

5. Share files between computers

During a digital investigation, we need to frequently upload data files to Forensics Workstation (herein Kali Linux VM). We can transfer files between the two computers by using a file transfer program, for example, WinSCP, a free and open source file transfer client for Microsoft Windows systems. However, for convenience, it would be better to make a shared folder between host and virtual machines in our circumstance. Note that your virtual machine running Kali Linux, aka the guest operating system, is a completely independent computer from the real computer running the host operating system (Windows in our examnple). Generally speaking, there are four different file sharing techniques including vmware-tools, samba, sftp and cloud storage. They all have different advantages and limitations, we will discuss them next.

1. File sharing by using vmware-tools

Vmware-tools is a suite of utilities that can give us more convenience by enhancing the performance of the virtual machine's operating system and improving management of the virtual machine. With vmware-tools, we can eliminate or improve these issues:

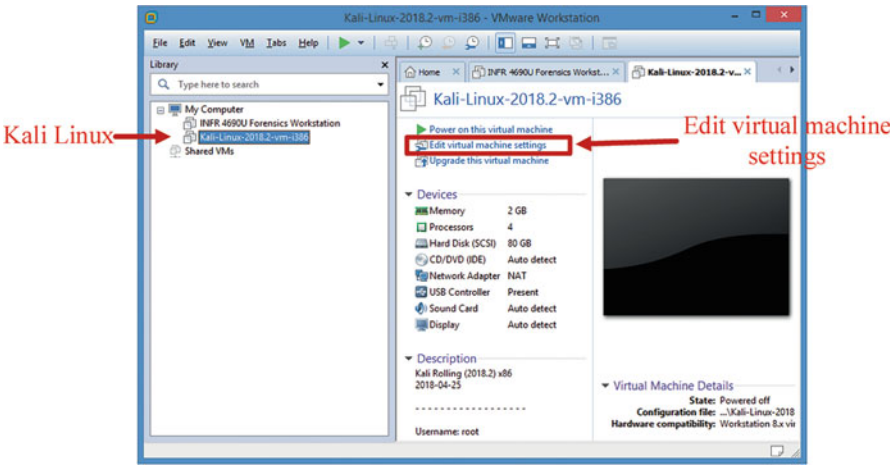
- Low video resolution
- Unable to resize your virtual machine
- Restrict movement of the mouse
- Inability to copy/paste and drag/drop files between host and virtual machine

Also, it enables us to create shared folders between virtual and host machines, which is a very fast way to share folders. However, it can only be used between vmware virtual machine and its host. When we have other virtual machine such as Virtualbox or even a real physical machine, VMware Tools is useless. The VMware Tools is installed by default in Kali Linux VM images. We can check the version of VMware Tools installed on Kali Linux VM by typing the following command

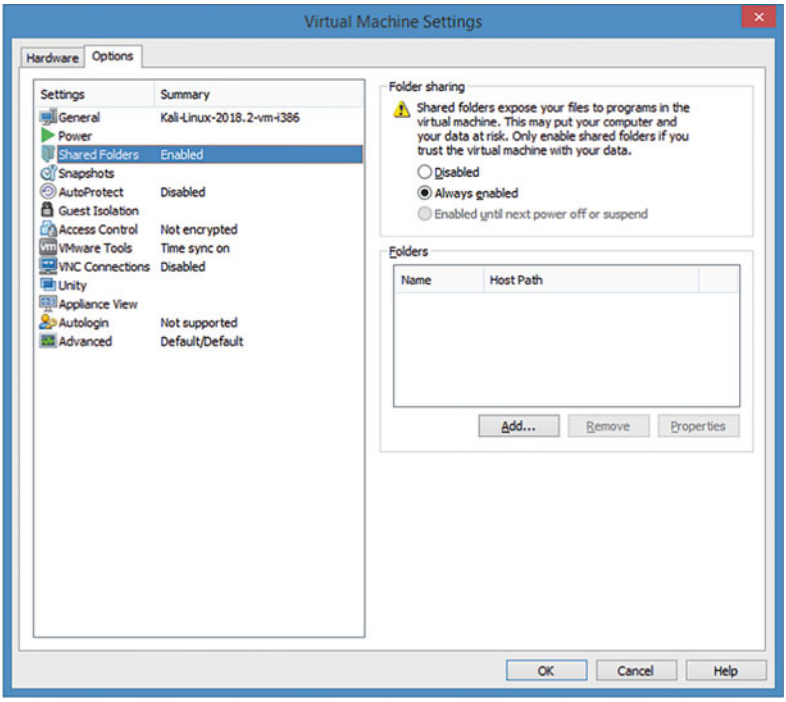
```
# vmware-toolbox-cmd -v
10.2.5.3619 (build-8068406)
```

After having VMware Tools installed on a VMware virtual machine, we can share files between virtual machine and host by creating shared folders. To create a shared folder, we must have VMware Tools correctly installed and then use the virtual machine control panel to specify the directories to be shared. Following is the detailed configuration of shared folder on the Kali Linux virtual machine.

(a) To set up one or more shared folders for a virtual machine, be sure the virtual machine is powered off. Click Edit the virtual machine settings

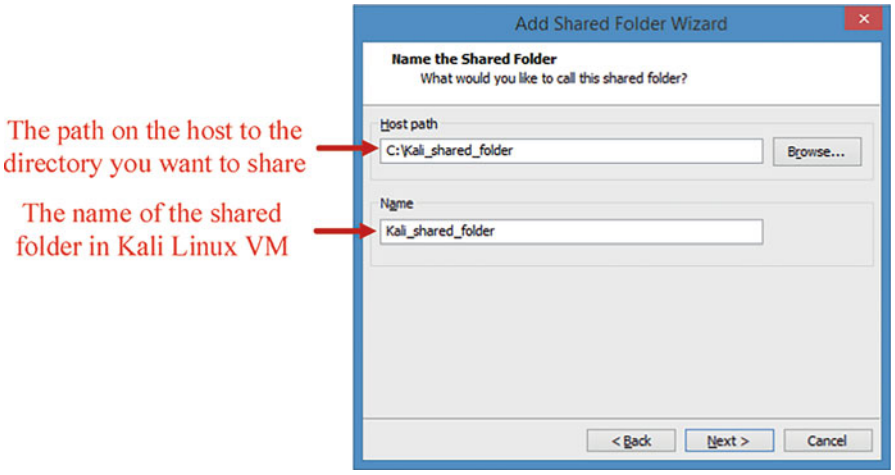



(b) Click Options->Shared Folders

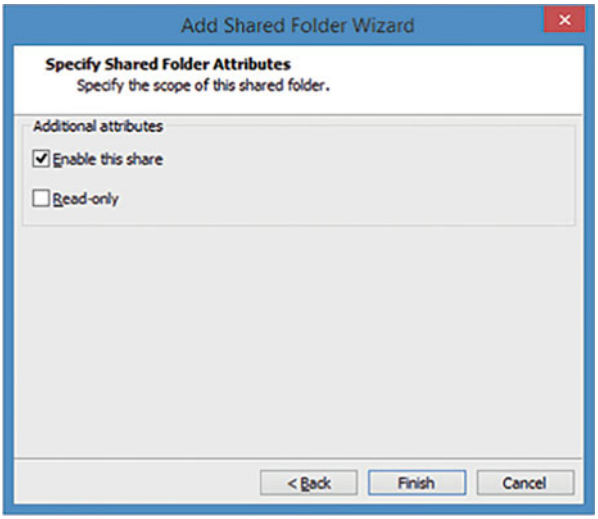



(c) Choose Always enabled for the folder sharing between virtual machine and host. Click Add to add a shared folder. The Add Shared Folder Wizard will guide you through the steps adding a new shared folder to Kali Linux VM.

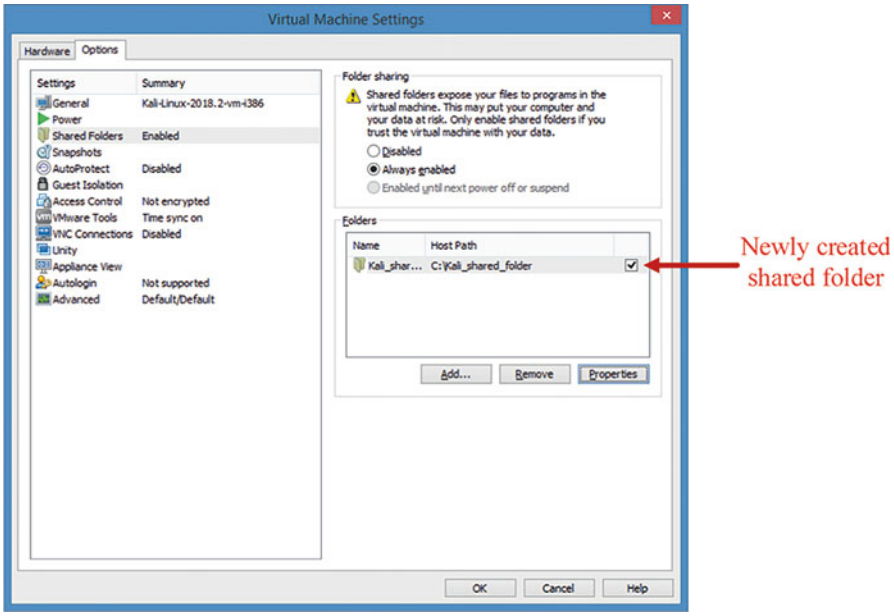
(d) Choose the path on the host to the directory you want to share. Type in the full path or browse to the directory




(e) Specify shared folder attributes and enable this share. Note that you can add a folder to the list without enabling it immediately. You can then enable the folder at any time by clicking its name in this list, clicking Properties and enabling the folder in the Properties dialog box.



(f) Click Finish to finish adding the shared folder.




It can be observed that the newly created shared folder appears in the list of shared folders. You can always select a shared folder and click Properties to change its attributes.

Note that you must run the mount-shared-folders.sh on the desktop (shown in Fig. 3.7) to mount Windows shared folder(s) to Kali Linux VM for them to be accessible in the folder of /mnt/hgfs after starting Kali Linux VM.

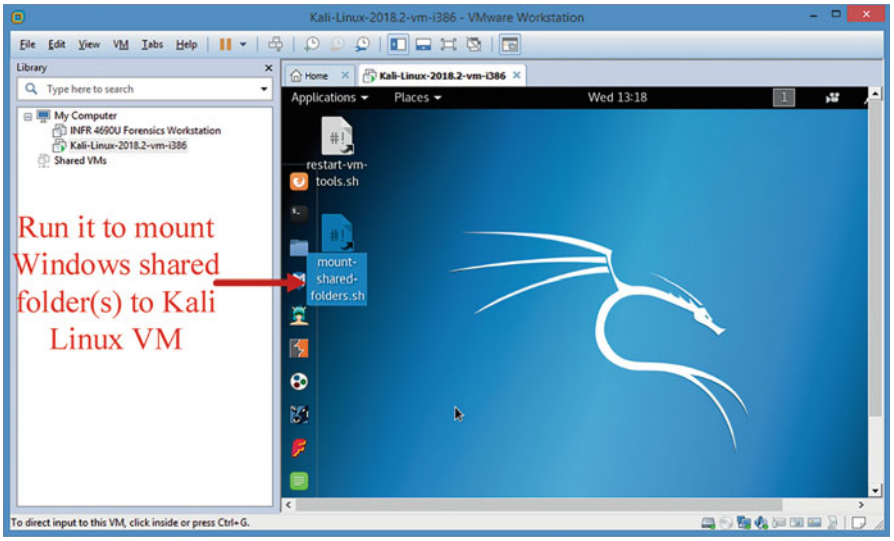
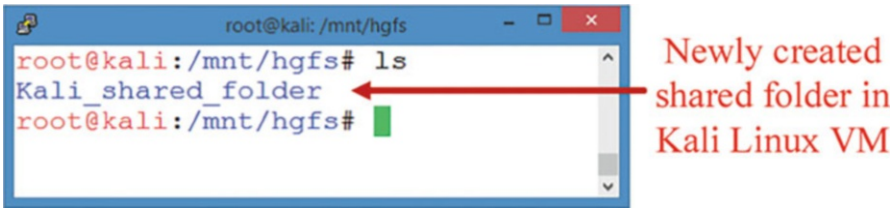



**Fig. 3.7** Mount shared folders in Kali Linux VM

(g) Once completed, the shared folder we created should be accessible in "/mnt/hgfs".




Afterwards, you can use newly created shared folder to share any type of file between your host machine and Kali Linux. However, windows shortcuts and Linux symbolic links do not work correctly if you try to share them via shared folders. Also, please do not open a file in a shared folder from more than one application at a time. For example, you should not open the same file using an application on the host operating system and another application in the guest operating system. In some circumstances, doing so could cause data corruption in the file.

2. File sharing by using Samba

Vmware-tools is convenient, but can only be used in VMware virtual machine environment, which is so circumscribed. As we know, Network File System (NFS) enables file sharing between Linux machines, whereas Common Internet File System (CIFS) helps us to share files between Windows machines. However, sharing files between Windows and Linux in a seamless way can be a little more complex. Next, we will show how to use Samba to create shared folders across both operating systems, which is applicable for more situations.

Samba is an implementation of SMB (Server Message Block) protocol, which is a file/resource protocol (Fig. 3.8). It facilitates file and printer sharing among Linux and Windows systems as an alternative to NFS. By using Samba, we have two ways to share files. First, by running Samba server in Linux, we can specify the shared folders and then gain access from windows; second, which is the opposite way, we can access windows shared folders from Linux by using Samba client. Next, we are going to describe a file sharing configuration based on Samba. Particularly, Kali Linux VM acts as Samba server and the host Windows machine works as Samba client.

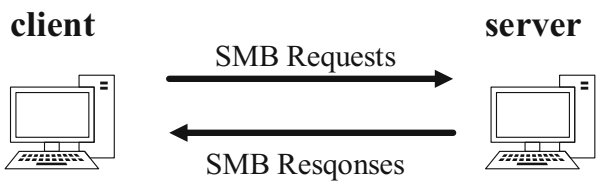


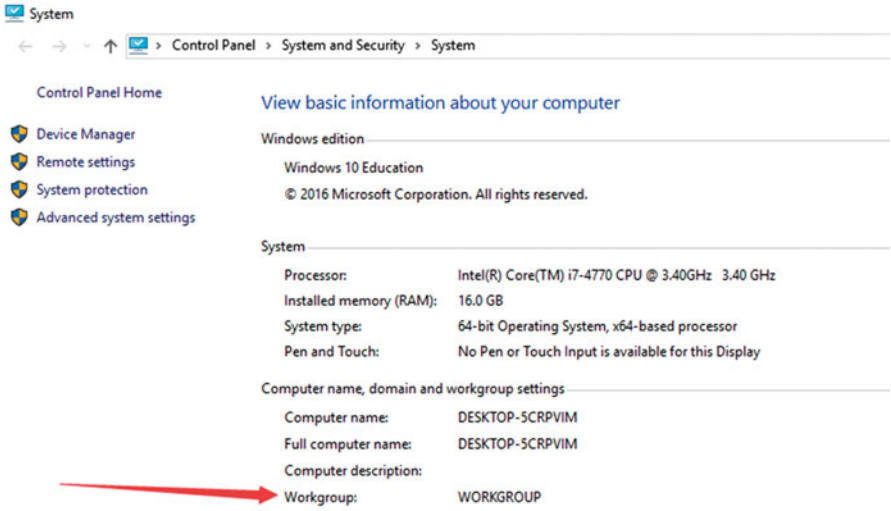

**Fig. 3.8** File sharing using Server Message Block (SMB)

**Build Samba server on Linux** To share files with Samba, we need to set up a Samba server in Kali Linux VM. We create shared folders on Linux machine which is running samba server. After that, we can visit the shared folders directly from windows file manager. Following is the samba Linux server setup step by step:

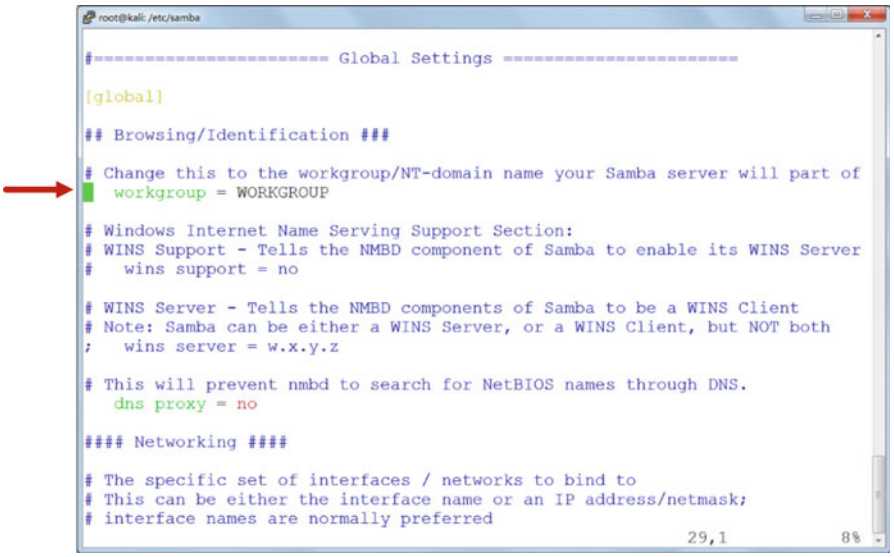
1. First, you should install Samba related software using the following command

```
# apt-get install samba
```

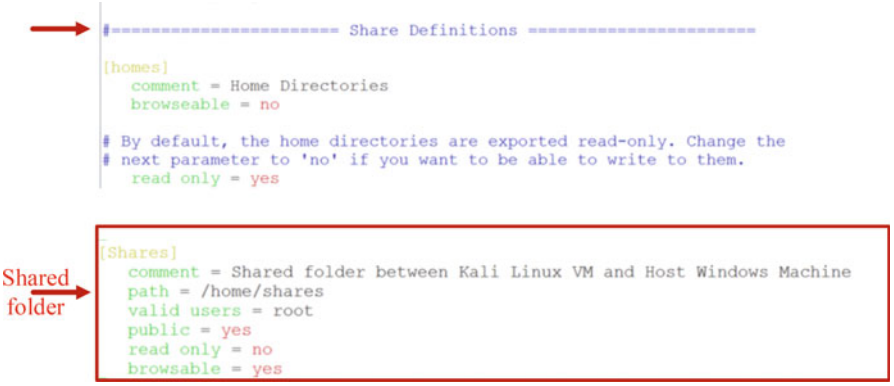
2. Then, the Samba configuration file can be found in its default system folder of /etc/samba/smb.cnf.
3. On your host Windows machine, navigate to the Control Panel. Click the System icon to find your Workgroup settings, including Workgroup name. In our example, we have "Workgroup = WORKGROUP".




4. To set up the share folders on Kali Linux machine, open the Samba configuration file and set the workgroup the same as windows workgroup. In our example, change the workgroup setting to WORKGROUP, shown below




5. Navigate to the "Share Definitions" section, and add a section named [Shares] like the followings for a shared folder between Kali Linux VM and host Windows machine




From the above setting, we create a shared folder "/home/shares" and authorized users who are able to access shared folder include "root". The "read only = no" means that authorized users can modify files within the shared folder from a Samba client (or herein host Windows machine). The "browsable = yes" indicates that all files in this path can be discovered by a Windows Samba client. It is worth noting that the shared folder "/home/shares" must exist in Kali Linux VM.

6. Change the owner and the group of the shared folder to “nobody” using the following command

```
# chown nobody:nobody /home/shares
```

7. To finish setting up newly created shared solder, add authorized user by the following command, and choose a password for this user when prompted. Note that the authorized user should be an existed user in Kali Linux system

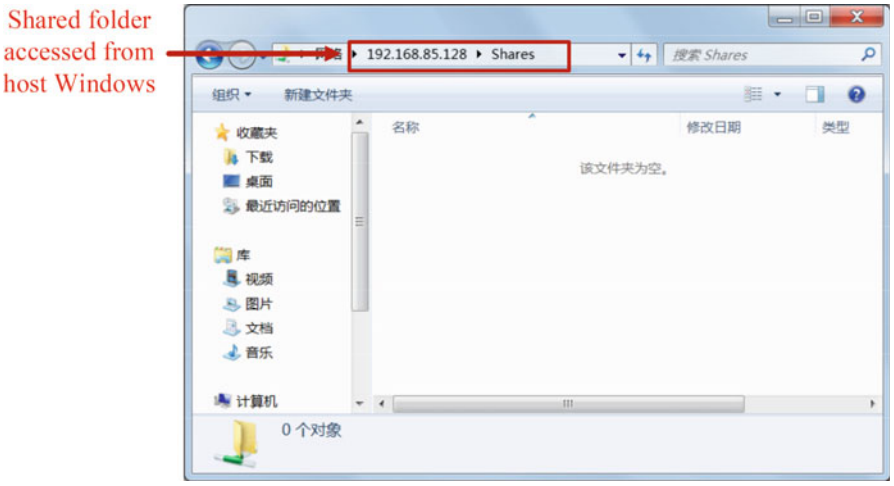
```
# pdbedit -a -u root
new password:
retype new password:
Unix username:       root
NT username:
Account Flags:       [U        ]
User SID:            S-1-5-21-4281320985-2316340312-3265071856-1000
Primary Group SID:   S-1-5-21-4281320985-2316340312-3265071856-513
Full Name:           root
Home Directory:      \\kali\root
HomeDir Drive:
Logon Script:
Profile Path:        \\kali\root\profile
Domain:              KALI
Account desc:
Workstations:
Munged dial:
Logon time:          0
Logoff time:         never
Kickoff time:        never
Password last set:   Sat, 09 Jun 2018 14:51:01 EDT
Password can change: Sat, 09 Jun 2018 14:51:01 EDT
Password must change: never
Last bad password   : 0
Bad password count  : 0
Logon hours         : FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
```

8. Once configuration is complete, we can start the Samba server using the following command

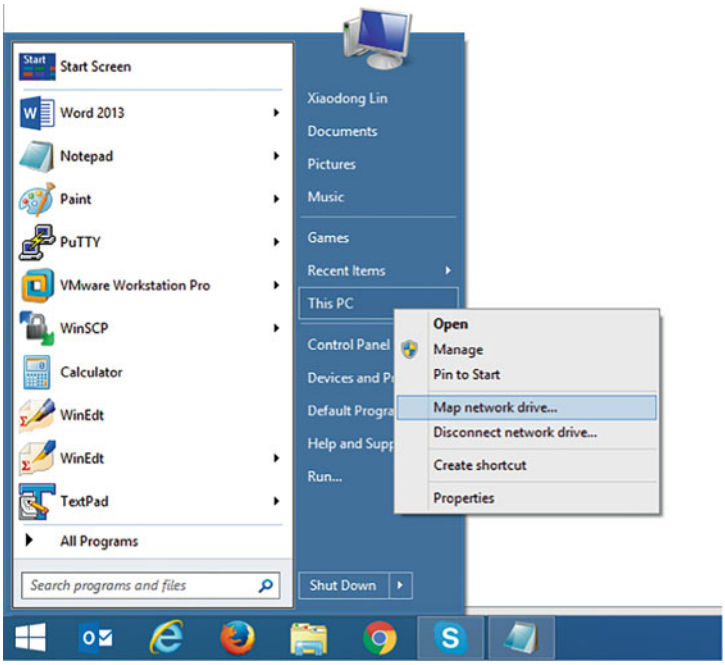
```
# smbd start
```

Note that you can test your modified Samba configuration file to verify its correctness by using the ‘testparm’ utility.

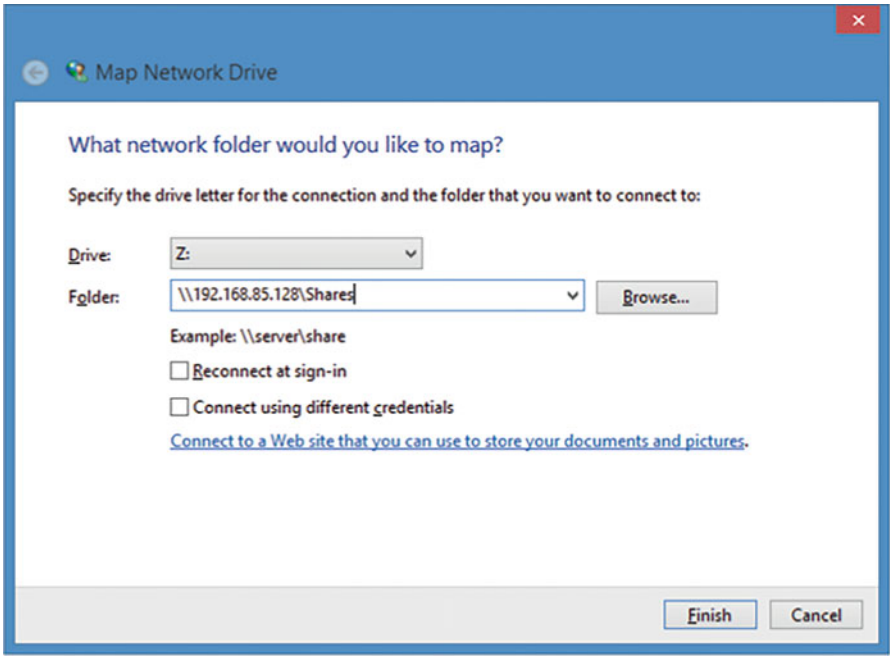
**Connect to Samba from Host Windows** Now, Samba server is running on Kali Linux and you can access the shared folders from host windows machine, which is a SMB client. Assume that the IP address of Kali Linux VM is 192.168.85.128. Open File Explorer in host Windows machine, enter \\192.168.85.128\Shares, and then type the user name "root" and the password you chosen into the popped up window to log in. Please be notified that the IP address of your Kali Linux VM may be different. You can use "ifconfig"utility to figure it out.




Or, you can set up a network drive by Opening the Start menu to Select "This PC". Then, Right-click on "This PC" and Select "Map network drive"



On the Map network drive dialog, select an available drive letter and enter\ \192.168.85.128\Shares into the Folder box. Click Finish.




Once it is complete, the shared network folder you just mapped should appear in a list of available Windows drives. In our example, the shared folder is mapped to drive Z.


3. File sharing by using ftp

The File Transfer Protocol (FTP) is a standard network protocol used to transfer files between a client and server. Most Linux machines already have vsftp installed, so we can easily transfer file with ftp. Also, for secure transfers, secure file transfer protocols are available. For example, we can use SFTP, which is a secure version of FTP protected by SSH.

First, in host Windows, we install winscp [9] or filezila [10], and then connect to the Kali Linux VM by typing the IP address, user name and password (example of

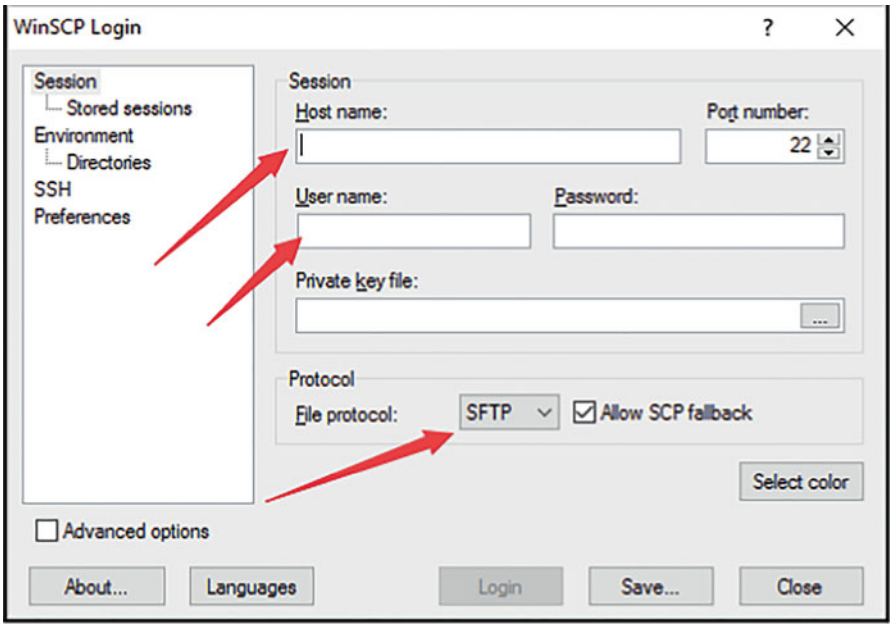



**Fig. 3.9** Transfer files using WinSCP

using winscp shown in Fig. 3.9). Afterwards, we can transfer files between Kali Linux VM and host Windows easily.

4. File sharing by using Dropbox-like cloud storage

Nowadays, cloud storage is also a convenient file sharing way. By enrolling in some cloud storage services like Dropbox, you can easily set up share folders between different computers. For more information about sharing file using Dropbox, please refer to https://www.dropbox.com/help/topics/sharing_files_and_folders.

## 3.4 First Forensic Examination Using TSK

Until now, you have successfully built your Forensics Workstation. Next, you can use the newly built Forensics Workstation to perform your first forensic examination. You will learn some of the most used TSK tools, and how they can be used. In doing so, sample image files were downloaded from the Computer Forensic Reference Data Sets (CFReDS) project website [11]. The Computer Forensic Reference Data Sets (CFReDS) for digital evidence is a repository of digital images developed by The National Institute of Standards and Technology (NIST) [12]. These data sets

were created as references, simulation and practice material for investigators hoping to further their digital forensic skills.

1. Download your forensic disk image from NIST government site using the following command

```
# wget http://www.cfreds.nist.gov/dfr-images/dfr-11-mft-ntfs.dd.bz2
```

Note that the downloaded test image is compressed with bzip2.

2. Extract disk image files using the following command

```
# bzip2 –d dfr-11-mft-ntfs.dd.bz2
```

The resulted disk image is dfr-11-mft-ntfs.dd, which is a test image used for the practice of deleted file recovery.

3. Use the mmls command to discover the layout of the disk image. With the mmls command we can find the image offset, or where the allocated partition starts.

```
# mmls -t dos dfr-11-mft-ntfs.dd
DOS Partition Table
Offset Sector: 0
Units are in 512-byte sectors

     Slot    Start        End          Length       Description
000:  Meta    0000000000   0000000000   0000000001   Primary Table (#0)
001:  -------  0000000000   0000000127   0000000128   Unallocated
002:  000:000  0000000128   0002091135   0002091008   NTFS / exFAT (0x07)
003:  -------  0002091136   0002097152   0000006017   Unallocated
```

where the "-t dos" option specifies the test image undertaking examination and testing is using DOS partitions, also known as PC-based Partitions.

In this example, we can clearly see that there is only one partition in the image. The locations of the starting sector and ending sector for the partition are Sector 128 and Sector 2091135, respectively. Thus, the size of the partition is 2091008 sectors.

4. Use the dcfldd command to extract the partition image from the disk image

```
# dcfldd if=dfr-11-mft-ntfs.dd bs=512 skip=128 count=2091008 of=ntfs.dd
```

where ntfs.dd is the name of the file used to store the extracted partition image. Please refer to **Appendix B** at the end of this chapter for detailed instructions on how to use dcfldd utility.

5. Use the fsstat command to display the details of the filesystem made on the partition

```
# fsstat -f ntfs ntfs.dd
FILE SYSTEM INFORMATION
--------------------------------------------
File System Type: NTFS
Volume Serial Number: 2ACADB0FCADAD5E3
OEM Name: NTFS
Volume Name: ntfs
Version: Windows XP

METADATA INFORMATION
--------------------------------------------
First Cluster of MFT: 43562
First Cluster of MFT Mirror: 65343
Size of MFT Entries: 1024 bytes
Size of Index Records: 4096 bytes
Range: 0 - 64
Root Directory: 5

CONTENT INFORMATION
--------------------------------------------
Sector Size: 512
Cluster Size: 8192
Total Cluster Range: 0 - 130686
Total Sector Range: 0 - 2091006

$AttrDef Attribute Values:
$STANDARD_INFORMATION (16)   Size: 48-72   Flags: Resident
$ATTRIBUTE_LIST (32)   Size: No Limit   Flags: Non-resident
$FILE_NAME (48)   Size: 68-578   Flags: Resident,Index
$OBJECT_ID (64)   Size: 0-256   Flags: Resident
$SECURITY_DESCRIPTOR (80)   Size: No Limit   Flags: Non-resident
$VOLUME_NAME (96)   Size: 2-256   Flags: Resident
$VOLUME_INFORMATION (112)   Size: 12-12   Flags: Resident
$DATA (128)   Size: No Limit   Flags:
$INDEX_ROOT (144)   Size: No Limit   Flags: Resident
$INDEX_ALLOCATION (160)   Size: No Limit   Flags: Non-resident
$BITMAP (176)   Size: No Limit   Flags: Non-resident
$REPARSE_POINT (192)   Size: 0-16384   Flags: Non-resident
$EA_INFORMATION (208)   Size: 8-8   Flags: Resident
$EA (224)   Size: 0-65536   Flags:
$LOGGED_UTILITY_STREAM (256)   Size: 0-65536   Flags: Non-resident
```

As mentioned earlier, the test disk image we used here is created for the purpose of practicing deleted file recovery forensic tools in TSK. Next, you will use TSK to recover deleted files found in the image. However, as an investigator, we are not

likely to know the names or contents of files deleted. Next we will use the **fls** command to find deleted files. You can use **man fls** from your system to learn more about available options with the manual provided.

6. Use the fls command to peruse the filesystem. We will be using the –r option to recursively move through directories. The option -o provides the offset number.

```
# fls -r ntfs.dd
```

7. Next, use the fls command to display only deleted files with the –d option.

```
# fls -r -d ntfs.dd
d/- * 0:    Orion
-/d * 36-144-1: Lyra
-/r * 41-128-1: Lyra/Sheliak.txt
-/r * 42-128-1: Lyra/Vega.txt
-/r * 43-128-1: Lyra/Sulafat.txt
```

We can see here that several text files were deleted. We will recover them next using the **icat** command, which is a TSK utility used to output the contents of a file based on its filesystem metadata (or the Master File Table (MFT) entry number in NTFS filesystem or inode number in extended file system (Ext) filesystem.

8. Recover deleted files using the icat command.

```
# icat -r ntfs.dd 41 > recovered_Sheliak.txt
# icat -r ntfs.dd 42 > recovered_Vega.txt
# icat -r ntfs.dd 43 > recovered_Sulafat.txt
```

Where the "-r" option specifies that icat uses file recovery techniques if the file is deleted. The numbers 41, 42 and 43 are the MFT entry numbers used by these deleted files, Sheliak.txt, Vega.txt, and Sulafat.txt, respectively. The details about NTFS filesystem will be covered in Chaps. 7 and 8. The recovered/deleted files are saved into files whose names start with a prefix "recovered_".

9. Finally, use the cat command to display your recovered files. Congratulations, you have successfully completed your first forensic examination by using TSK to recover deleted files.

```
# cat recovered_Sheliak.txt

DFR
File Sheliak.txt path Lyra
++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++++++++++++++++++++++++++++++++++++++++++++++++
Sheliak.txt
```

**Review Questions**

1. What does NFS stand for?
2. What is the default URL used by a web browser connecting to the Autopsy service?
3. What is the relationship between TSK and Autopsy?
4. What utility in The Sleuth Kit (TSK) displays the layout of a disk?

   (a) mmls
   (b) fsstat
   (c) blkcat
   (d) xxd

5. Samba is a network file sharing protocol.

   (a) True
   (b) False

## 3.5 Practice Exercise

The objective of this exercise is to learn how to conduct a digital forensics investigation using Kali. Particularly, you are required to analyze a disk image using TSK and Autopsy, from creating cases in Autopsy to practicing various forensic analysis techniques in TSK and Autopsy.

### 3.5.1 Setting Up the Exercise Environment

For this exercise, you will use a disk image named "thumbimage_fat.dd", provided in the book. It can be found in the diskimages subfolder. You will need to upload this disk image to Forensics Workstation you have built earlier in this chapter. Note that you need to remember the location where you upload the disk image file "thumbimage_fat.dd" since this information is required when you add the disk image for analysis in Autopsy.

### 3.5.2 Exercises

**Part A: Starting Your Autopsy Forensic Browser**

- Start your Forensics Workstation (or Kali Linux VM) and Login as root onto it.
- Start Autopsy and launch Firefox to access its web interface using the URL of http://localhost:9999/autopsy.

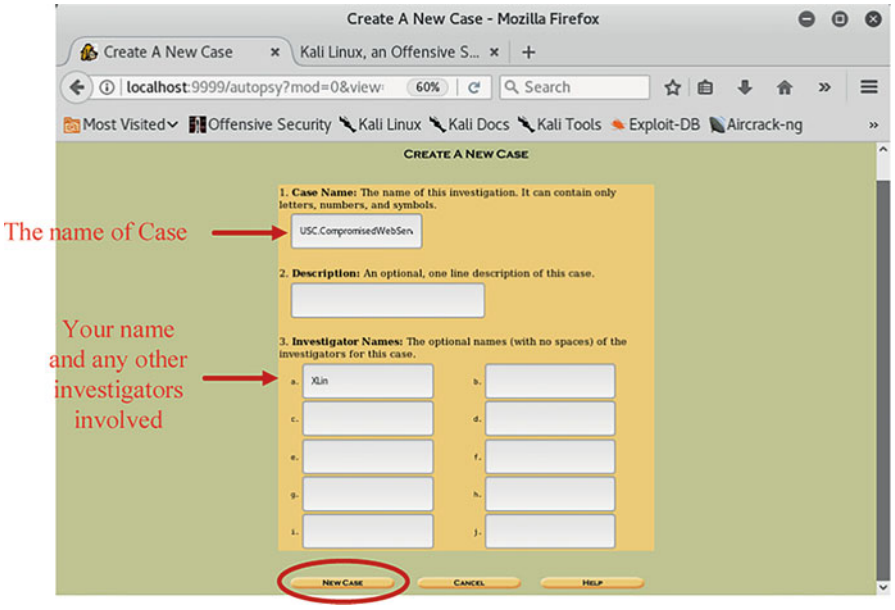
**Part B: Starting a New Case in Autopsy**

You already started the Autopsy Forensic Browser, and the default start page should be displayed as shown in Fig. 3.10. Now, you can start your investigation by creating a new case in Autopsy


**Fig. 3.10** Create a new investigation case in Autopsy

- Click New Case

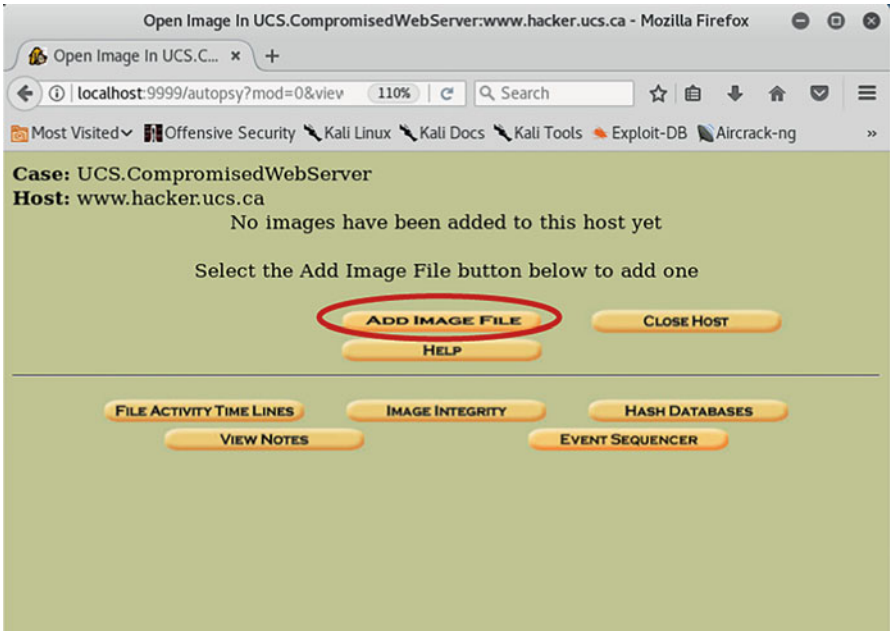
  Note that you will need enter all the necessary details when you create your investigation case. Assume that you are called in to investigate a computer compromise occurred at the University of Cyber Security, which is located in Waterloo, Ontario, Canada. The host name of the compromised Web server is www.hacker.ucs.ca, and "thumbimage_fat.dd" used in the exercise is the disk image you acquired at the crime scene.
- Enter the case details and Click New Case to continue. Note that the name of the Case must contain information which can be used to identify cases. In our example, the name of the Case could be "UCS.CompromisedWebServer".



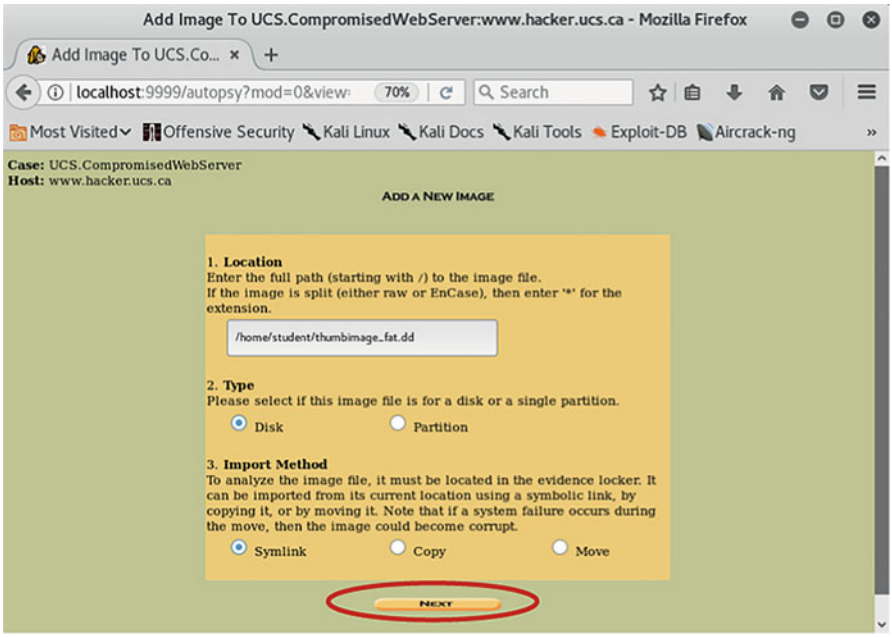

- Click ADD HOST Button.

- Enter the host details on the "ADD A NEW HOST" page and Click ADD HOST to continue.
- Click ADD IMAGE Button to add the image file of the added host for analysis.


- On the following page about the added host, click "ADD IMAGE FILE" to continue.



- Enter the image file details on the ADD A NEW IMAGE page and Click Next Button. In the example, we upload the image file into /home/student. Enter the path of the image file, /home/student/thumbimage_fat.dd, in the Location field. Since this image file is from a disk, select the "Disk" radio button. Also, there are three import methods available, select the "Symlink" radio button.




- The next page shows the details of the imported image. On the Image File Details page, select the "Calculate the hash value for the image" radio button and click ADD to continue.
- The MD5 hash value will be printed out. Be sure to write down the MD5 hash value of the image calculated by Autopsy and click OK to continue.

Now, you have successfully created an investigation case, and a default investigation page should be displayed in Fig. 3.11. Now you can analyze the digital evidence (or disk image) in Autopsy by clicking ANALYZE Button to try a variety of evidence analysis techniques, for example, keyword search.

Q1. What is the MD5 hash value of the disk image "thumbimage_fat.dd" calculated in Autopsy?

**Part C: Using Autopsy for Forensic Disk Analysis**

After you click ANALYZE Button in Fig. 3.11, the following interface appears with a list of tabs on the top of the screen. Each tab stands for an evidence search technique, except for HELP and CLOSE. Note that the list of evidence search

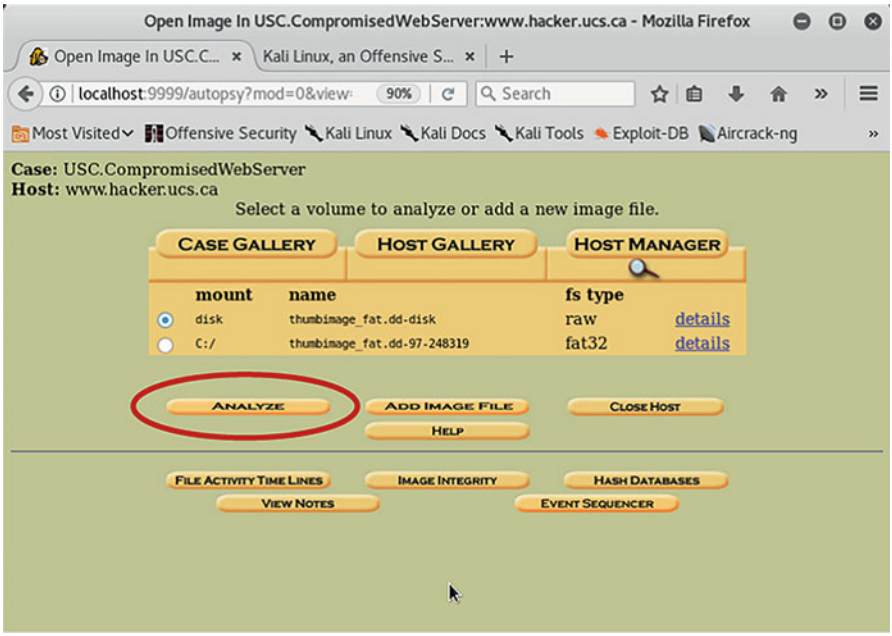



**Fig. 3.11** Digital Investigation Analysis in Autopsy

functionalities can be variably used, depending on which type of data (disk or file system image) for analysis. In Fig. 3.12, only a limited set of functionalities and interfaces, including keyword search, image details and data unit analysis, are enabled since we chose to analyze a disk image in Fig. 3.11.

We can clearly see in Fig. 3.12 that Autopsy provides a list of evidence search functionalities [13]:

- File analysis: This technique helps Autopsy analyze files and directories as well as the names of deleted files and Unicode-based file names.
- Keyword search: This technique allows Autopsy to configure keyword searches, of file system image, that can be performed using ASCII strings and grep regular expressions. Faster searches can be created for index files and strings that are searched frequently, can be configured into Autopsy for automated searching.
- File type analysis: This technique allows Autopsy to identify files based on their contents and internal structures. It can also be used to find hidden files.
- Image details: This technique allows Autopsy to view file system details as well as, on-disk layout and times of activity. This will provide information useful, during data recovery.
- Meta data analysis: This technique allows Autopsy to analyze Metadata structures that contain details on Files and Directories. This is useful if a deleted content in a file needs to be recovered. To do this, Autopsy will search directories so full path of the file can be identified where the structure is allocated.

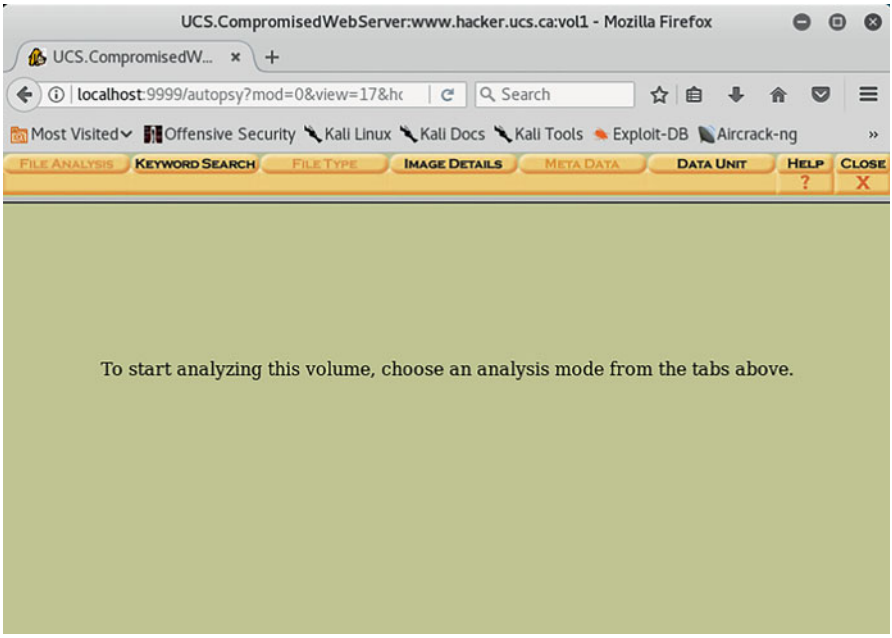



**Fig. 3.12** Digital evidence search techniques in Autopsy

- Data unit analysis: This technique allows Autopsy to analyze data units of the stored file content. It allows you to view the contents of any data unit, in ASCII, hex dump, and strings. Autopsy will search the Metadata structures, with the file type, in order to identify, which file has allocated the data unit.

Among these evidence search techniques mentioned above, keyword searching is one of the most common forensic techniques. In the early stages of a digital investigation, it is very typical that investigators don't have any leads in a case, but do know some specific keywords of interest to the investigation, for example, "forensics", "pornography", etc. Then, the investigators can develop their hypothesis to continue their investigation. In this exercise, you are asked to search a keyword "**Wikipedia**" in the disk image provided. After the disk image has been searched, a list of "hits" will appear on the left-hand side. Each data unit that contains the string is listed with the offset of each occurrence.

In Fig. 3.12, click on the Keyword search tab to complete this part of the exercise and answer the following questions.


Note that for this exercise, you need to make keyword search case insensitive.

Q2. How many hits when the search keyword is encoded to ASCII format?

Q3. What is the number (or address) of the data unit where the keyword resides?

Fundamentally, computers deal with numbers, particularly binary digits. When storing letters and other characters, they assign a number for each one. In other words, these letters and characters must be encoded in a way used to uniquely identify them, where ASCII, which stands for American Standard Code for Information Interchange (ISO 14962:1997), is the most common technique for encoding the characters. ASCII is a way of assigning specific 8-bit strings (a string of 0s and 1s of length 8) to the alphanumeric characters and punctuation. ASCII uses only 1 byte per character and a 1 byte scheme can only represent 256 symbols. However, there are many languages in the world, with their own alphabets or with their own accented versions of the ASCII romanized alphabets. Obviously, 8 bits per character is not sufficient. This is why multiple byte character encoding standards were developed. A very popular 2 byte (16 bit) encoding standard is called "Unicode", which can represent 65,000+ characters (two to the power of 16). In other words, Unicode is able to encompass the characters of all the world's living languages

Q4. How many hits when the search keyword is encoded as Unicode?

## Appendix A Installing software in Linux

You will need to become root (or superuser) to install software.

There are many ways to install software in Linux, and it can be accomplished either graphically or using the command line. There are two popular ways of installing software in Linux, installing software from source code and installing software with Apt [14], a Linux package manager for Debian and Debain-based Linux distributions like Ubuntu and Kali Linux.

**Note that there exist many Linux distributions, and the way of how to install software is slightly different for each distribution. Kali Linux is based on Debian Linux, which uses Apt.**

### (a) Using the "apt-get" commands to manage packages in Linux

**apt-get** Apt-get performs installations, package searches, updates and many other operations to software packages available to your Debian and Debain-based Linux systems.

For example, to install a package, use:

```
% apt-get install [package_name]
```

To remove a package, use:

```
% apt-get remove [package_name]
```

### (b) Compiling and installing software from source in Linux

The installation procedure for a software that comes in tar.gz (or tgz) and tar.bz2 packages isn't always the same, but usually it's like the following, assuming that the name of the package containing the source code of the program is archive:

| | |
|---|---|
| # tar -zxvf archive.tar.gz (or tar -zxvf archive.tgz) or tar -xvjf archive.tar.bz2 | Decompress the files contained in the zipped and tarred archive called archive |
| # cd archive | Change directory to software package |
| # ./configure | Execute the script preparing the installed files for compiling, including Makefile |
| # make | GNU make utility to maintain groups of programs |
| # make install | Install the software |

## Appendix B dcfldd Cheat Sheet

dcfldd is "an enhanced version of GNU dd with features useful for forensics and security", for example, creating a forensic image of an entire disk. The basic syntax of the command is:

dcfldd if= input file bs=512 skip=0 count=1 of= output file

This command will read data from the source (drive or file) and write that to an output file (or drive). It will then read one block from the beginning of the input file. The block size for transferring has been set to 512 bytes.

Where:

1. If indicates input file. Example input files include:

LINUX

| | |
|---|---|
| File name | The input file |
| /dev/stdin | "standard input" (stdin) device, i.e., keyboard |
| /dev/hda | (First IDE Physical Drive) |
| /dev/hda2 | (Second Logical Partition) |
| /dev/sda | (First SCSI Physical Drive) |

WINDOWS

| | |
|---|---|
| File name | The input file |
| \\.\PhysicalDrive0 | (First Physical Drive) |
| \\.\D: | (Logical Drive D:) |
| \\.\PhysicalMemory | (Physical Memory) |

2. Of indicates output file. Example output files include

| | |
|---|---|
| imagefile.img | (Bit Image File) |
| /dev/usb | (USB Drive) |
| /dev/hdb | (2nd IDE Drive) |

3. Useful Options

| | |
|---|---|
| bs=block size | (Sets the block size) |
| count=N | (Copy only N blocks of input file) |
| skip=N | (Skip ahead N blocks FILE. By default, skip=0, which means it reads input file from beginning.) |

| | |
|---|---|
| conv=noerror,sync | (Do not skip on errors) |
| hashwindow=num | (Hash every num bytes) |
| hashwindow=0 | (Hash entire file) |
| hashlog=filename | (Write md5 hash to file) |

4. Usages and Examples

   (a) Create a disk image

   Example: dcfldd if=/dev/sdb of=/datatraveller.img

   This command will create a disk image of external USB drive, and write the image to an output file called datatraveller.img.

   (b) Wipe out hard drives and flash drives, for example, with all zero

   Example: dcfldd if=/dev/zero of=/dev/sdb

   This command will fill external USB drive with zeros.

   (c) Extract a random portion of a data file

   Example: dcfldd if=thumbimage_fat.dd bs=512 skip=0 count=1 of=mbr.dd

   Assume that thumbimage_fat.dd is an image of MBR disk. This command will extract the MBR of the disk.

## References


1. B. Carrier, "The Sleuth Kit," 2017. [Online]. Available: www.sleuthkit.org.
2. http://www.porcupine.org/forensics/tct.html
3. https://www.symantec.com/connect/articles/freeware-forensics-tools-unix
4. C. Marko. Introduction to The Sleuth Kit (TSK). 2005.
5. Sleuthkit.org, "Sleuth Kit Wiki," Sleuthkit, [Online]. Available: https://wiki.sleuthkit.org/index.php?title=Main_Page. [Accessed February 2017].
6. Autopsy. https://www.sleuthkit.org/autopsy/desc.php
7. The SANS Investigative Forensic Toolkit (SIFT). https://digital-forensics.sans.org/community/downloads
8. What is Virtualization? https://www.igi-global.com/dictionary/an-evolutionary-approach-for-load-balancing-in-cloud-computing/31852
9. https://winscp.net/eng/download.php
10. https://filezilla-project.org/
11. The Computer Forensic Reference Data Sets (CFReDS) Project. [Online]. Available: http://www.cfreds.nist.gov/
12. https://www.nist.gov/
13. https://digital-forensics.sans.org/blog/2009/05/11/a-step-by-step-introduction-to-using-the-autopsy-forensic-browser
14. A Beginners Guide to using apt-get commands in Linux(Ubuntu). https://codeburst.io/a-beginners-guide-to-using-apt-get-commands-in-linux-ubuntu-d5f102a56fc4