



CySA+ Lab Series

Lab 11: Digital Forensic Analysis

Document Version: **2022-10-10**

Material in this Lab Aligns to the Following	
CompTIA CySA+ (CS0-002) Exam Objectives	4.3 - Given an incident, analyze potential indicators of compromise 4.4 - Given a scenario, utilize basic digital forensics techniques 5.2 - Given a scenario, apply security concepts in support of organizational risk mitigation
All-In-One CompTIA CySA+ Second Edition ISBN-13: 978-1260464306 Chapters	17: Analyze Potential Indicators of Compromise 18: Utilize Basic Digital Forensics Techniques 20: Security Concepts in Support of Organizational Risk Mitigation

Copyright © 2022 Network Development Group, Inc.
www.netdevgroup.com

NETLAB+ is a registered trademark of Network Development Group, Inc.
KALI LINUX™ is a trademark of Offensive Security.
ALIEN VAULT OSSIM V is a trademark of AlienVault, Inc.
Microsoft®, Windows®, and Windows Server® are trademarks of the Microsoft group of companies.
Greenbone is a trademark of Greenbone Networks GmbH.
SECURITY ONION is a trademark of Security Onion Solutions LLC.
Android is a trademark of Google LLC.
pfSense® is a registered mark owned by Electric Sheep Fencing LLC ("ESF").
All trademarks, logos, and brand names are the property of their respective owners.

Contents

Introduction	3
Objective	3
Lab Topology	4
Lab Settings	5
1 Analyzing Disk Image Files with The Sleuth Kit.....	6
2 Analyzing Image Files with Autopsy.....	14

Introduction

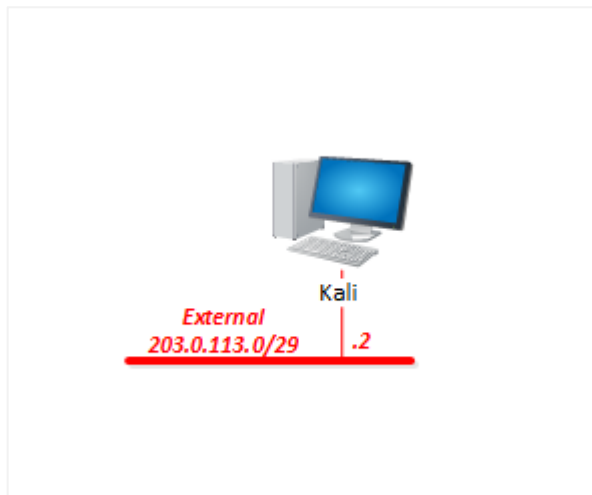
Computer forensics is defined specifically as the “*the application of investigation and analysis techniques to gather and preserve evidence from a particular computing device in a way that is suitable for presentation in a court of law*” [<https://searchsecurity.techtarget.com/definition/computer-forensics>]. Even if a Cybersecurity Analyst is not collecting information for legal purposes, they still need to perform forensic audits of computers and other devices.

In this lab, you will explore the forensic capabilities of *The Sleuth Kit*, and its GUI component, *Autopsy*. Forensic investigation is a critical part of the incident response process, helping you to analyze an image of a partition without relying on a live client.

Objective

- Discuss image creation using dd
- Use The Sleuth Kit to examine an image and recover deleted files
- Use Autopsy to examine an image and recover deleted files
- Learn how to extract files hidden in images with steganography

Lab Topology



Lab Settings

The information in the table below will be needed in order to complete the lab. The task sections below provide details on the use of this information.

Virtual Machine	IP Address	Account	Password
WinOS (Server 2019)	192.168.0.50	Administrator	NDGlabpass123!
MintOS (Linux Mint)	192.168.0.60	sysadmin	NDGlabpass123!
OSSIM (Alien Vault)	172.16.1.2	root	NDGlabpass123!
UbuntuSRV (Ubuntu Server)	172.16.1.10	sysadmin	NDGlabpass123!
Kali	203.0.113.2	sysadmin	NDGlabpass123!
pfSense	203.0.113.1 172.16.1.1 192.168.0.1	admin	NDGlabpass123!

1 Analyzing Disk Image Files with The Sleuth Kit

Creating an image file is a vital part of using The Sleuth Kit or Autopsy. It is important to create the image as close to the incident as possible, to preserve the contents of the volatile memory and any deleted files that may exist before the space is overwritten.

While this lab will not actually create an image, as this requires creating and destroying partitions without external media, it is still an important function to understand. Linux has the built-in capability to create an image using the *dd* command.

The *dd* command has multiple functions, but there are two ways to best utilize the tool for forensic investigation. The first is to completely clone a partition into a blank partition, which would use a format such as:

`dd if=<partition location to be cloned> of=<empty partition location>.`

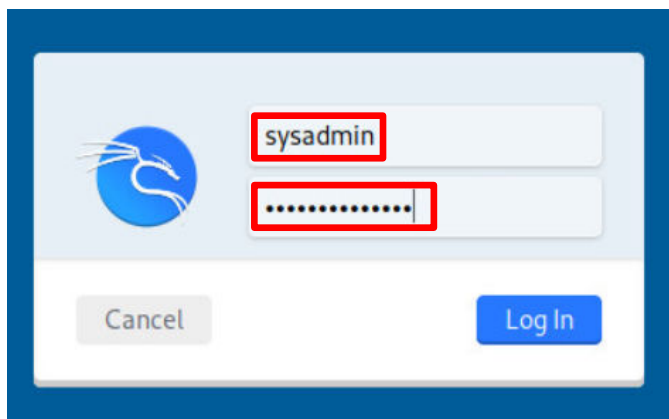
The second is to create an image using a format such as:

`dd if=<partition location to be cloned> of=<directory location/filename.dd>`

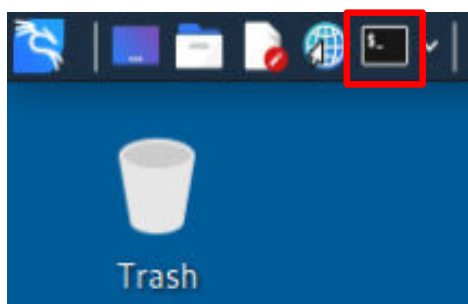
The latter of these methods has been utilized to provide the basic images you will use for this lab.

In this task, you will use *The Sleuth Kit* commands to analyze the sleuthkit.dd image and recover a deleted file named sesame.txt.

1. Set the focus to the **Kali** computer.
2. Log in as **sysadmin** using the password: NDGLabpass123!

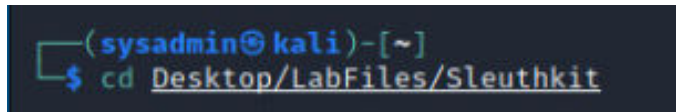


3. Click the **Terminal** icon to open a terminal window.



4. Navigate to the **Sleuthkit** directory with the following command:

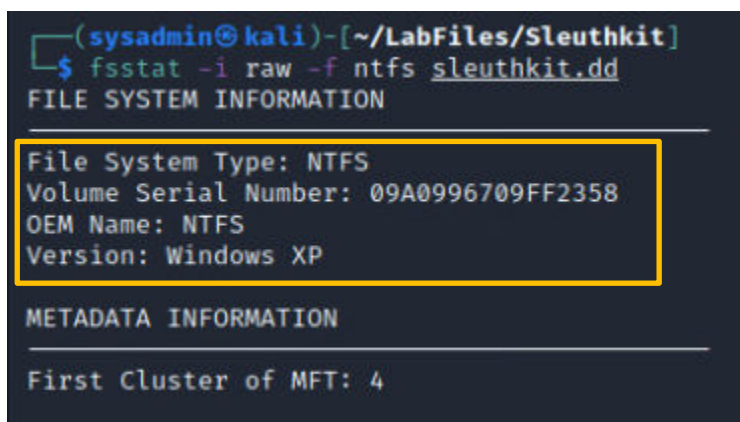
```
cd Desktop/LabFiles/Sleuthkit
```



```
(sysadmin@kali)-[~]  
$ cd Desktop/LabFiles/Sleuthkit
```

5. Use the following command to view information about the **sleuthkit.dd** image. This image was extracted from a machine utilizing the **NTFS** file system, along with other system information available to you.

```
fsstat -i raw -f ntfs sleuthkit.dd
```



```
(sysadmin@kali)-[~/LabFiles/Sleuthkit]  
$ fsstat -i raw -f ntfs sleuthkit.dd  
FILE SYSTEM INFORMATION  
-----  
File System Type: NTFS  
Volume Serial Number: 09A0996709FF2358  
OEM Name: NTFS  
Version: Windows XP  
METADATA INFORMATION  
-----  
First Cluster of MFT: 4
```



The **dd** file extension is used for low-level compression disk images.

6. To view the raw files contained within the image, type the following command:

```
fls -i raw -f ntfs sleuthkit.dd
```

```
(sysadmin@kali)-[~/LabFiles/Sleuthkit]
$ fls -i raw -f ntfs sleuthkit.dd
r/r 4-128-1: $AttrDef
r/r 8-128-2: $BadClus
r/r 8-128-1: $BadClus:$Bad
r/r 6-128-1: $Bitmap
r/r 7-128-1: $Boot
d/d 11-144-2: $Extend
r/r 2-128-1: $LogFile
r/r 0-128-1: $MFT
r/r 1-128-1: $MFTMirr
r/r 9-128-2: $Secure:$SDS
r/r 9-144-3: $Secure:$SDH
r/r 9-144-4: $Secure:$SII
r/r 10-128-1: $UpCase
r/r 10-128-2: $UpCase:$Info
r/r 3-128-3: $Volume
d/d 66-144-2: .Trash-0
r/r 64-128-2: open.jpg
r/- * 0:      sesame.txt
V/V 71: $OrphanFiles
```



There is a picture file located in this image (open.jpg), along with a deleted file (sesame.txt). Deleted files are marked with an asterisk, and may or may not be recoverable.

Make a note of the first number in the second column for the **open.jpg** file, **64** in this case. This is the **inode** number that will be used in the next step.



Inodes are utilized by *Linux/Unix* systems to uniquely identify files or directories and contain metadata about these objects.

7. Using the **inode** number for **open.jpg** gathered in the previous step, you can view information about the **open.jpg** file, including its type, size, and creation date. Additionally, you can view the sectors where the actual data is stored. Type the following command:

```
istat -i raw -f ntfs sleuthkit.dd 64
```

```
(sysadmin@kali)-[~/LabFiles/Sleuthkit]
$ istat -i raw -f ntfs sleuthkit.dd 64
MFT Entry Header Values:
Entry: 64          Sequence: 1
$LogFile Sequence Number: 0
Allocated File
Links: 1

$STANDARD_INFORMATION Attribute Values:
Flags: Archive
Owner ID: 0
Security ID: 0 ( )
Created:          2019-03-13 08:17:40.103297500 (EDT)
File Modified:    2019-02-26 05:40:58.000000000 (EST)
MFT Modified:     2019-03-13 08:17:40.130489400 (EDT)
Accessed:         2019-03-13 08:17:40.103297000 (EDT)

$FILE_NAME Attribute Values:
Flags: Archive
Name: open.jpg
Parent MFT Entry: 5      Sequence: 5
Allocated Size: 561152   Actual Size: 0
Created:          2019-03-13 08:17:40.103297500 (EDT)
File Modified:    2019-03-13 08:17:40.103297500 (EDT)
MFT Modified:     2019-03-13 08:17:40.103297500 (EDT)
Accessed:         2019-03-13 08:17:40.103297500 (EDT)

Attributes:
Type: $STANDARD_INFORMATION (16-0)  Name: N/A  Resident  size: 48
Type: $FILE_NAME (48-3)  Name: N/A  Resident  size: 82
```

8. Using the **inode** number, you can extract the **open.jpg** file using the following command:

```
icat -i raw -f ntfs sleuthkit.dd 64 > open.jpg
```

```
(sysadmin@kali)-[~/LabFiles/Sleuthkit]
$ icat -i raw -f ntfs sleuthkit.dd 64 > open.jpg
```

9. Issuing the **ls** command will show that the file **open.jpg** has been extracted in the present working directory.

```
ls
```

```
(sysadmin@kali)-[~/LabFiles/Sleuthkit]
$ ls
autopsy.dd  open.jpg  sleuthkit.dd
```

10. To gather some basic information about the **open.jpg** file, type the following command:

```
file open.jpg
```

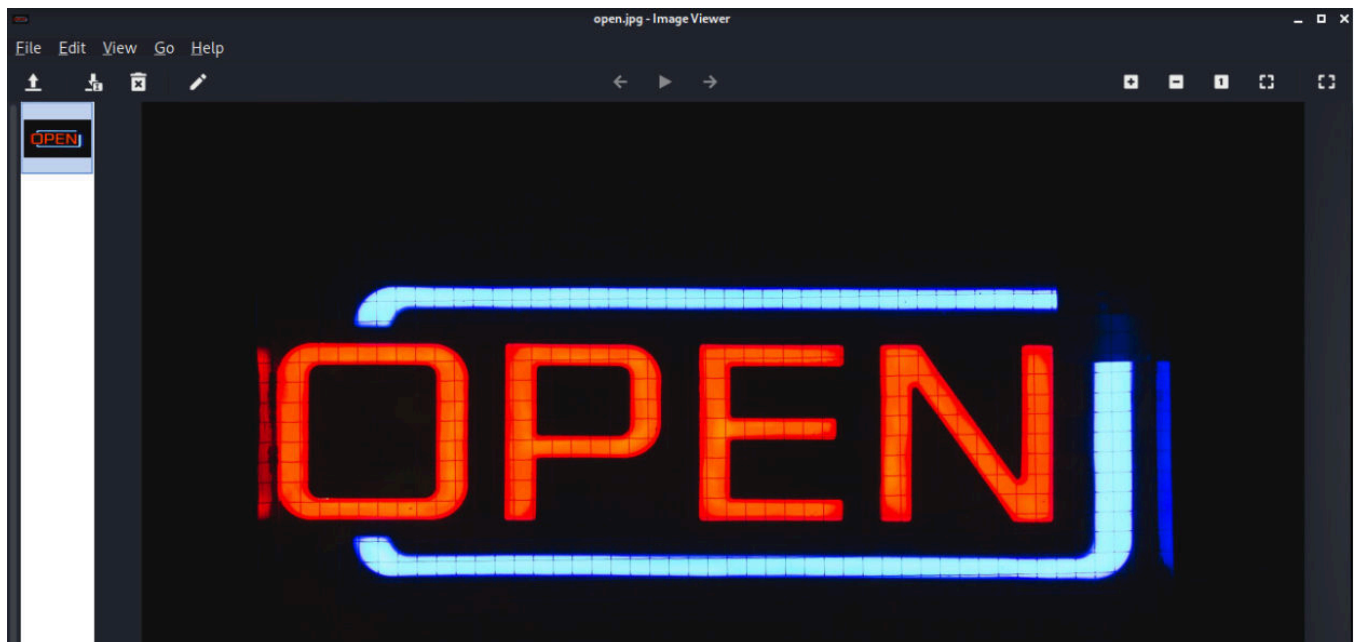
```
(sysadmin@kali)-[~/LabFiles/Sleuthkit]
$ file open.jpg
open.jpg: JPEG image data, JFIF standard 1.01, resolution (DPI), density 72x72, segment length 16, Exif Standard: [TIFF image data, big-endian, direntries=1, orientation=upper-left], baseline, precision 8, 3889x2187, components 3
```

11. To view the contents of the extracted picture file, use the following command.

```
ristretto open.jpg
```

```
(sysadmin@kali)-[~/LabFiles/Sleuthkit]
$ ristretto open.jpg
```

Image Viewer will open and show the **open.jpg** file which was extracted from the disk image.



12. Close *ImageMagick* by clicking the **X** in the upper-right of the screen.



You may have noticed that while the deleted file **sesame.txt** is visible, it does not have an *inode* number associated with it.

This means you cannot simply extract it from the image in the same manner that you did with the **open.jpg** file.

13. Now, recover the deleted **sesame.txt** file. Since you have no **inode** number for this file, you will need to recover all the deleted files on the image. Type the following command:

```
tsk_recover -i raw -f ntfs -v -e sleuthkit.dd recovery
```

```
(sysadmin@kali)-[~/LabFiles/Sleuthkit]
$ tsk_recover -i raw -f ntfs -v -e sleuthkit.dd recovery
tsk_img_open: Type: 1 NumImg: 1 Img1: sleuthkit.dd
tsk_img_findFiles: sleuthkit.dd found
tsk_img_findFiles: 1 total segments found
raw_open: segment: 0 size: 2146435072 max offset: 2146435072 path: sleuthkit.dd
raw_read: byte offset: 0 len: 65536
raw_read: found in image 0 relative offset: 0 len: 65536
raw_read_segment: opening file into slot 0: sleuthkit.dd
```

At the bottom of the output, you will see there are 3 files recovered.

```
ntfs_proc_attrseq: Processing extended entry for primary entry 23
ntfs_proc_attrseq: Resident Attribute in Type: 16 Id: 0 IdNew: 0 Name:
Files Recovered: 3
```

14. Once the command has finished, issue the **ls** command to view that there is a new folder named **recovery**.

```
ls
```

```
(sysadmin@kali)-[~/LabFiles/Sleuthkit]
$ ls
autopsy.dd  open.jpg  recovery  sleuthkit.dd
```

15. Navigate into the **recovery** directory using the following command:

```
cd recovery
```

```
(sysadmin@kali)-[~/LabFiles/Sleuthkit]
$ cd recovery

(sysadmin@kali)-[~/LabFiles/Sleuthkit/recovery]
$
```

16. Issue another **ls** with the **-a** switch to view hidden files.

```
ls -a
```

```
(sysadmin@kali)-[~/LabFiles/Sleuthkit/recovery]
$ ls -a
.  ..  open.jpg  .Trash-0
```

17. From here, you can see that the **.Trash-0** directory, which is hidden, was recovered as well. Navigate to it with the following command:

```
cd .Trash-0
```

```
(sysadmin@kali)-[~/LabFiles/Sleuthkit/recovery]
$ cd .Trash-0
```

18. To view the contents of the present working directory, Type the **ls** command:

```
ls
```

```
(sysadmin@kali)-[~/LabFiles/Sleuthkit/recovery/.Trash-0]
$ ls
files  info
```

19. Two new directories are shown; navigate into the **files** directory with the following command:

```
cd files
```

```
(sysadmin@kali)-[~/LabFiles/Sleuthkit/recovery/.Trash-0]
$ cd files
```

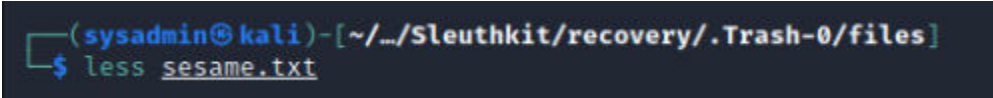
20. Issue another **ls** command. Note that this reveals the same **sesame.txt** file that lacked an **inode** in your image analysis.

```
ls
```

```
(sysadmin@kali)-[~/Sleuthkit/recovery/.Trash-0/files]
$ ls
sesame.txt
```

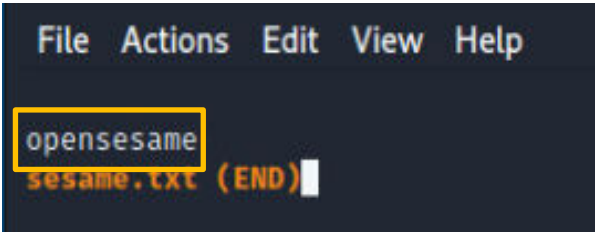
21. View the **sesame.txt** file with the following command:

```
less sesame.txt
```



```
(sysadmin@kali)-[~/Sleuthkit/recovery/.Trash-0/files]  
$ less sesame.txt
```

22. Looking at the contents of **sesame.txt**, you notice it contains a single word, **opensesame**. While the purpose of this file and its contents are unknown at this point, you think that it might be useful later, so you make a note of it. Press **q** to close the file content display.



```
File Actions Edit View Help  
opensesame  
sesame.txt (END)
```

23. Remain on the *Kali* computer and continue on to the next task.

2 Analyzing Image Files with Autopsy

In this scenario, you are investigating a device that has recently experienced a security breach. To contain the damage, it is often best practice to create an image of the compromised machine to isolate and contain any further damage or loss of data. To investigate, you will follow a similar process to view information from a different image (autopsy.dd) using Autopsy (Sleuth Kit's GUI). Additionally, you will extract a text file from the autopsy.dd image that contains a "virus payload" that could have been responsible for the breach.

1. At the prompt, type the following command to open the forensic browser.

```
sudo autopsy
```

If asked for the **sysadmin** password, type: NDGLabpass123!

```
(sysadmin@kali)-[~]
$ sudo autopsy

Autopsy Forensic Browser
http://www.sleuthkit.org/autopsy/
ver 2.24

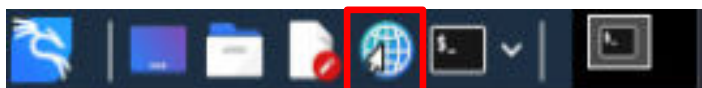
Evidence Locker: /var/lib/autopsy
Start Time: Sat Oct  2 12:56:09 2021
Remote Host: localhost
Local Port: 9999

Open an HTML browser on the remote host and paste this URL in it:

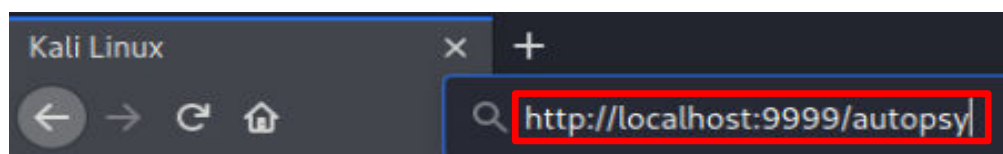
http://localhost:9999/autopsy

Keep this process running and use <ctrl-c> to exit
```

2. With the *Autopsy Forensic Browser* process running, open a browser window by clicking the **Firefox** icon.



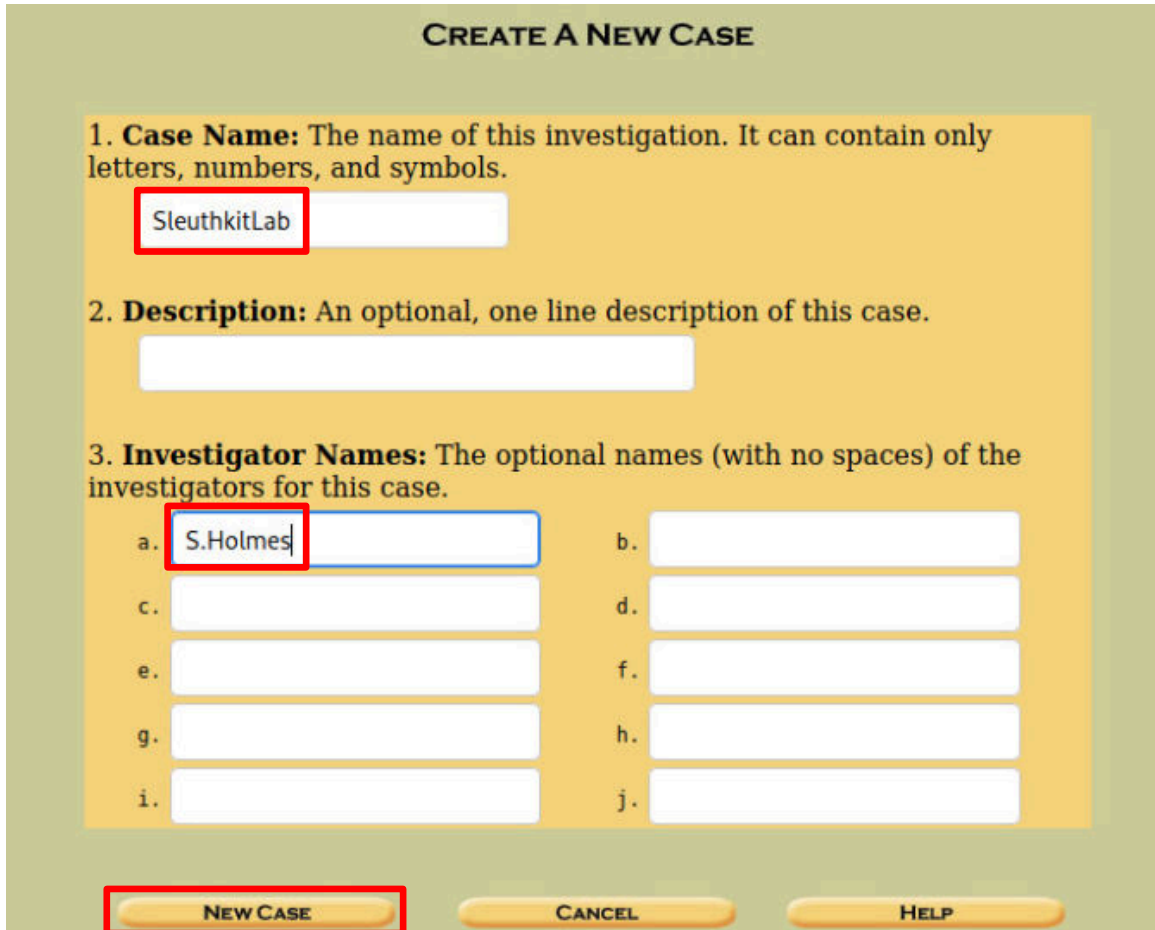
3. In the address bar of the browser, go to the following address: `http://localhost:9999/autopsy`.



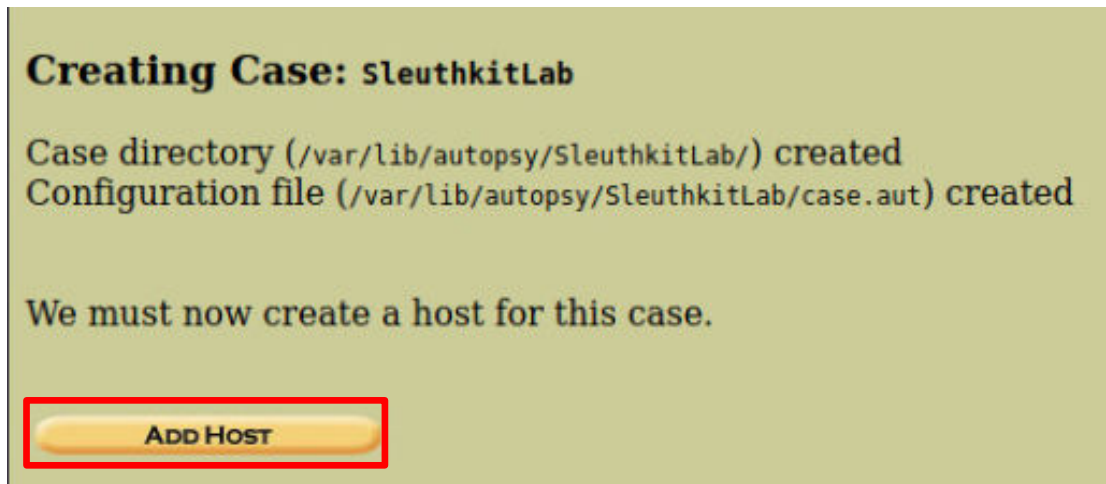
4. Once in the *Autopsy Forensic Browser*, click on **New Case**.



5. On the *Create a New Case* window, type SleuthkitLab as the *Case Name*, and your name as an *Investigator Name*. Then, click **New Case**.



6. On the *Creating Case: SleuthkitLab* screen, click **Add Host**.



7. On the *Add a New Host* screen, leave all fields at their default values and click **Add Host**.

ADD A NEW HOST

1. **Host Name:** The name of the computer being investigated. It can contain only letters, numbers, and symbols.

2. **Description:** An optional one-line description or note about this computer.

3. **Time zone:** An optional timezone value (i.e. EST5EDT). If not given, it defaults to the local setting. A list of time zones can be found in the help files.

4. **Timeskew Adjustment:** An optional value to describe how many seconds this computer's clock was out of sync. For example, if the computer was 10 seconds fast, then enter -10 to compensate.

5. **Path of Alert Hash Database:** An optional hash database of known bad files.

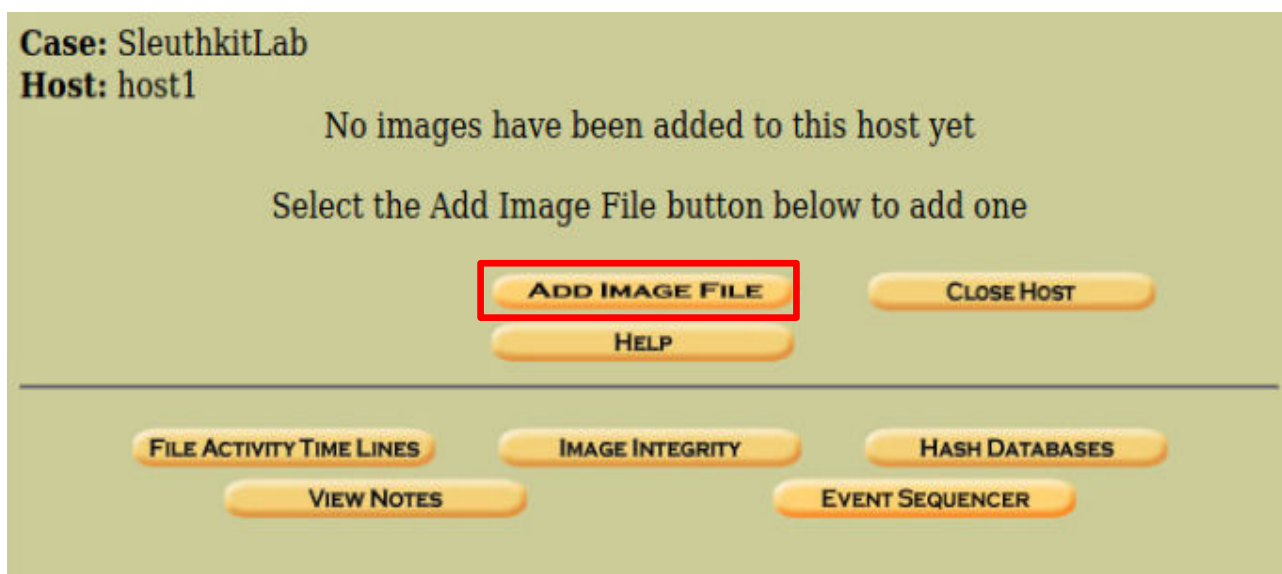
6. **Path of Ignore Hash Database:** An optional hash database of known good files.

ADD HOST **CANCEL** **HELP**

8. On the *Adding host: host1 to case SleuthkitLab* screen, click **Add Image**.



9. Now you must select an image. Do so by clicking **Add Image File**.



10. On the *Add a New Image* screen, type `/home/sysadmin/Desktop/LabFiles/Sleuthkit/autopsy.dd` in the *Location* field. Under *Type*, click **Partition**, under *Import Method* leave **Symlink** checked, and then click **Next**.

Case: SleuthkitLab
Host: host1

ADD A NEW IMAGE

1. Location
Enter the full path (starting with /) to the image file.
If the image is split (either raw or EnCase), then enter '*' for the extension.

2. Type
Please select if this image file is for a disk or a single partition.

☐ Disk ☒ Partition

3. Import Method
To analyze the image file, it must be located in the evidence locker. It can be imported from its current location using a symbolic link, by copying it, or by moving it. Note that if a system failure occurs during the move, then the image could become corrupt.

☒ Symlink ☐ Copy ☐ Move

NEXT

11. On the *Image File Details* screen, leave all the defaults and then click **Add**.

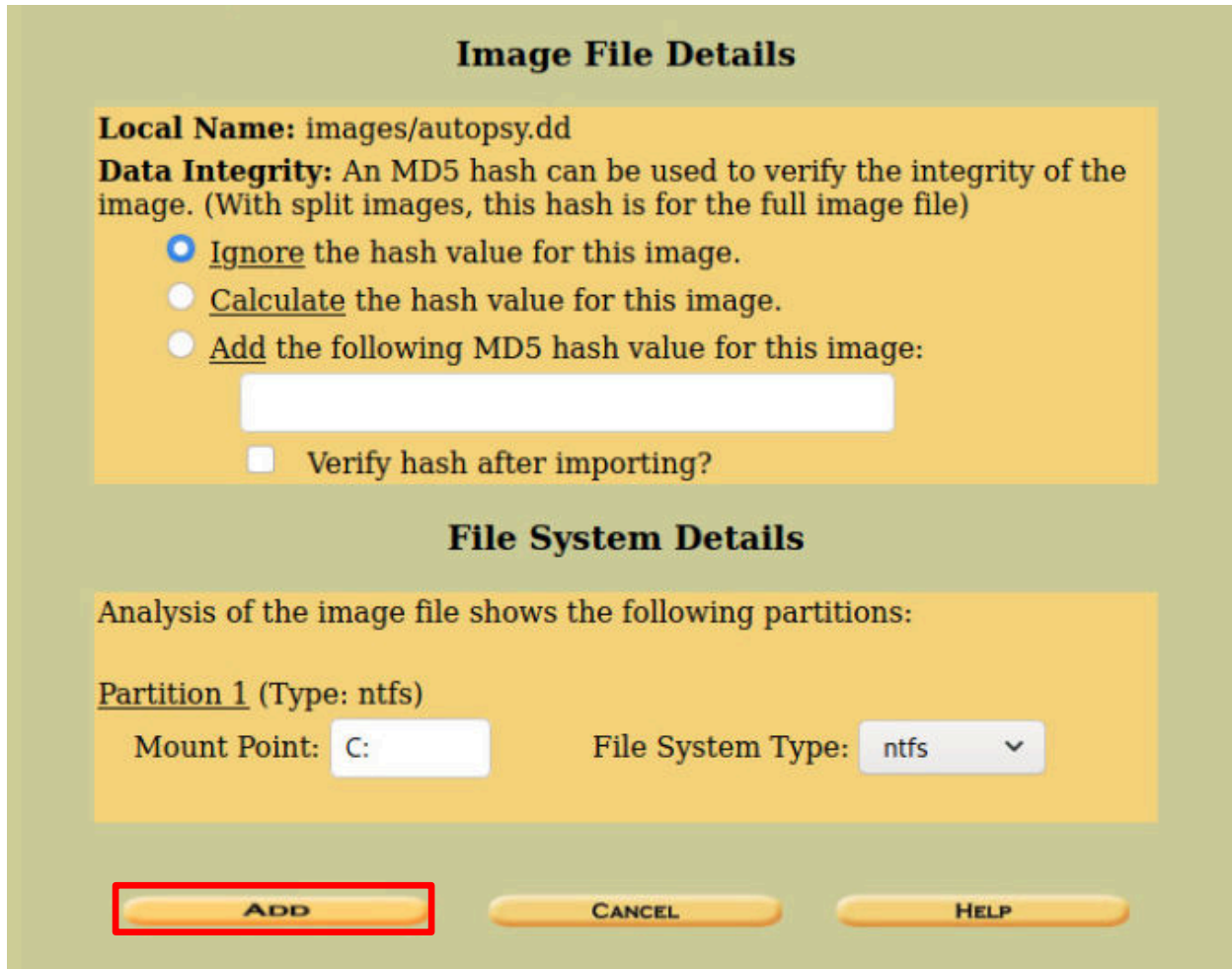


Image File Details

Local Name: images/autopsy.dd

Data Integrity: An MD5 hash can be used to verify the integrity of the image. (With split images, this hash is for the full image file)

- ☒ Ignore the hash value for this image.
- ☐ Calculate the hash value for this image.
- ☐ Add the following MD5 hash value for this image:

☐ Verify hash after importing?

File System Details

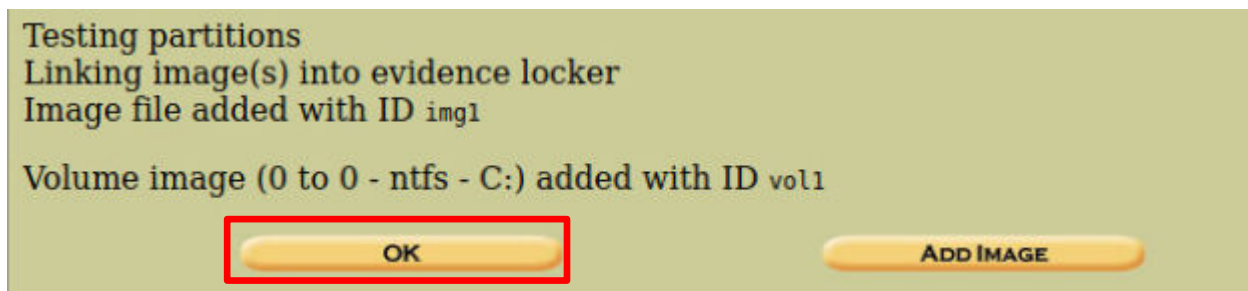
Analysis of the image file shows the following partitions:

Partition 1 (Type: ntfs)

Mount Point: File System Type:

ADD **CANCEL** **HELP**

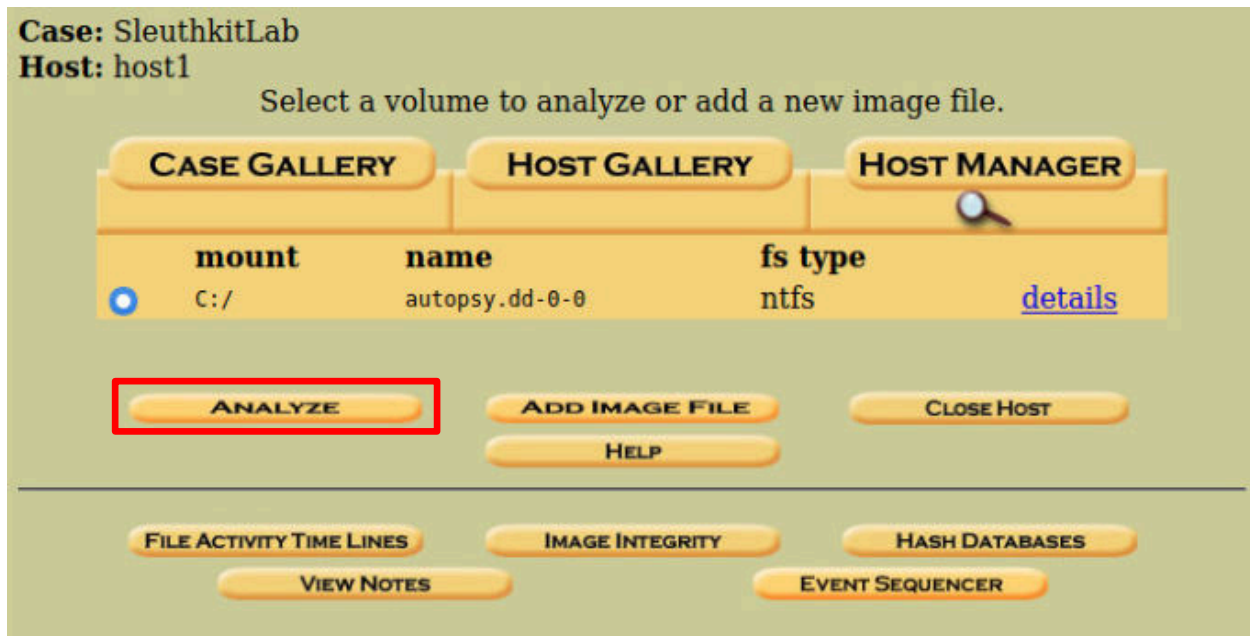
12. On the *Testing Partitions* screen, click **OK** to confirm.



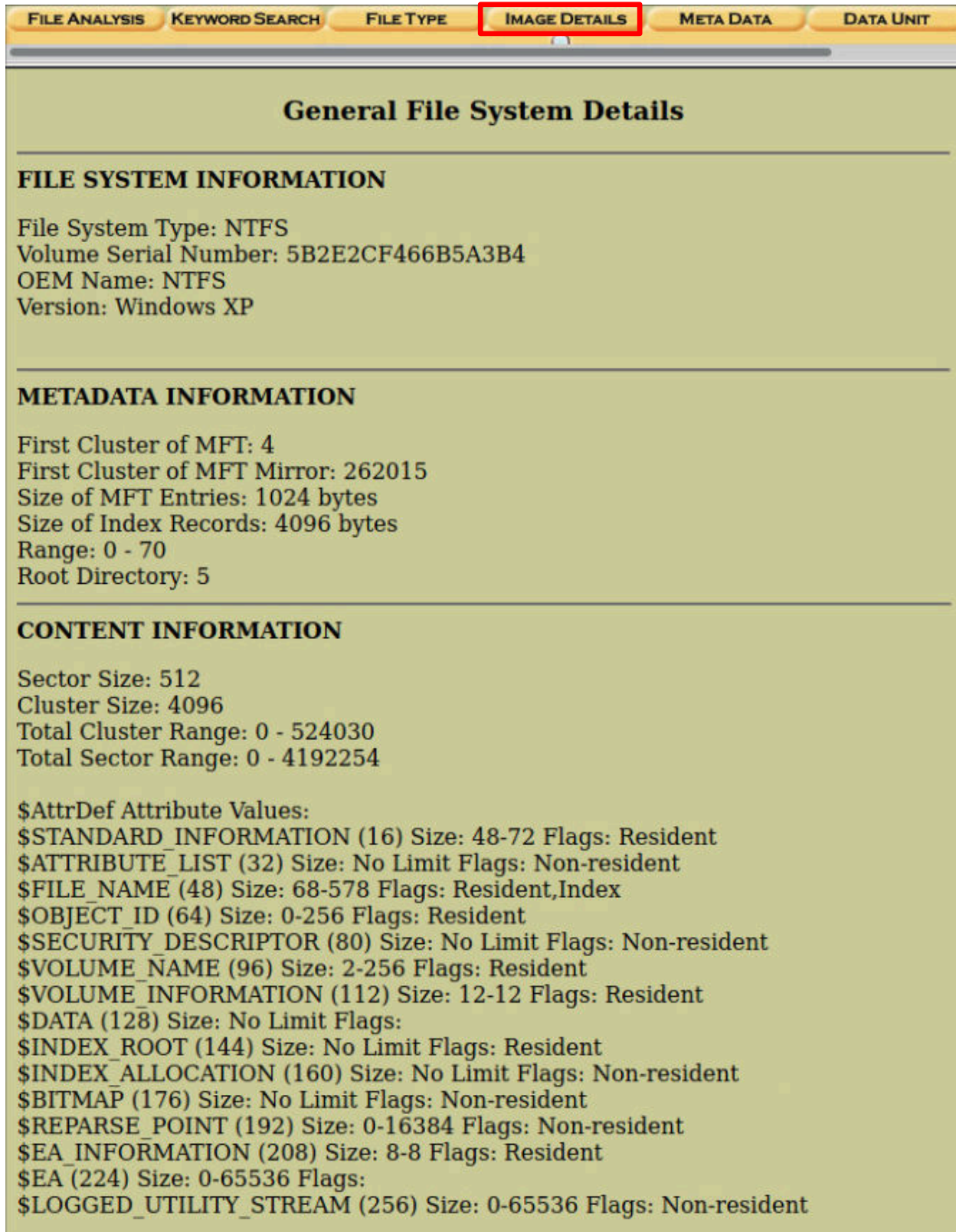
Testing partitions
Linking image(s) into evidence locker
Image file added with ID img1
Volume image (0 to 0 - ntfs - C:) added with ID vol1

OK **ADD IMAGE**

13. Confirm that the information is correct, as shown on the screen below, before clicking **Analyze**.



14. Now that the image is loaded, click on **Image Details** at the top of the page. Notice that several details are shown on this screen, similar to the CLI.



FILE ANALYSIS **KEYWORD SEARCH** **FILE TYPE** **IMAGE DETAILS** **META DATA** **DATA UNIT**

General File System Details

FILE SYSTEM INFORMATION

File System Type: NTFS
Volume Serial Number: 5B2E2CF466B5A3B4
OEM Name: NTFS
Version: Windows XP

METADATA INFORMATION

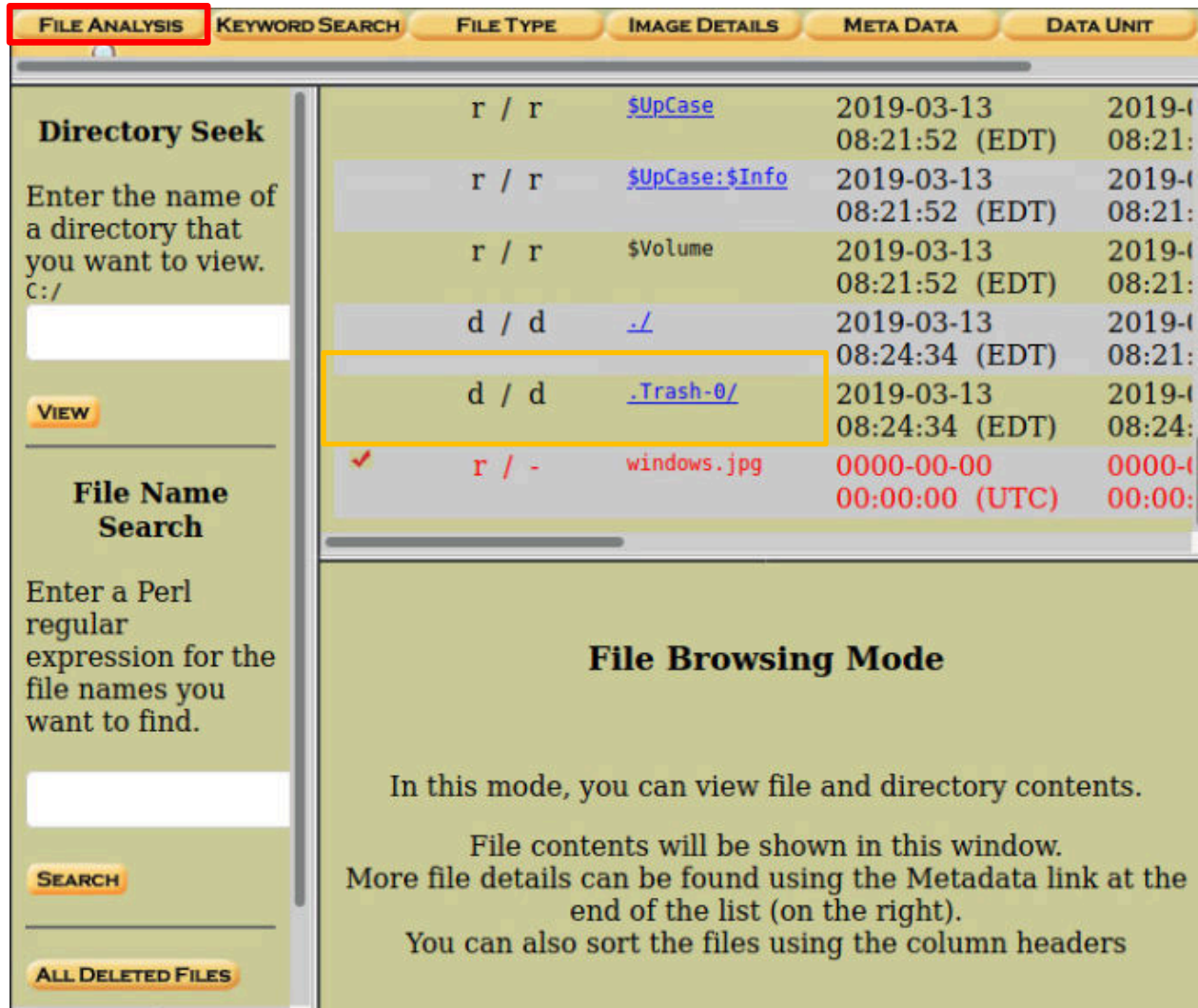
First Cluster of MFT: 4
First Cluster of MFT Mirror: 262015
Size of MFT Entries: 1024 bytes
Size of Index Records: 4096 bytes
Range: 0 - 70
Root Directory: 5

CONTENT INFORMATION

Sector Size: 512
Cluster Size: 4096
Total Cluster Range: 0 - 524030
Total Sector Range: 0 - 4192254

\$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

15. Click on **File Analysis** at the top of the page, and in the upper-right pane (it will show *Current Directory C:*) scroll down to the bottom of the pane, and you will not see the deleted file *sesame.txt* since this deleted file lacks an *inode* number to properly look it up and extract it. However, the **.Trash-0** folder exists in this image.



The screenshot shows the File Analysis interface with the 'FILE ANALYSIS' tab selected. The 'Directory Seek' pane on the left contains a search box and buttons for 'VIEW', 'File Name Search', 'SEARCH', and 'ALL DELETED FILES'. The main pane displays a list of files and directories. The following table represents the data shown in the list:

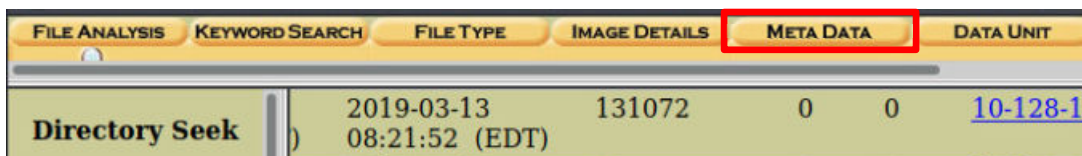
Permissions	File Name	Size	Modified	MD5	SHA1	SHA256
r / r	\$UpCase	2019-03-13 08:21:52 (EDT)	2019-03-13 08:21:52 (EDT)			
r / r	\$UpCase:\$Info	2019-03-13 08:21:52 (EDT)	2019-03-13 08:21:52 (EDT)			
r / r	\$Volume	2019-03-13 08:21:52 (EDT)	2019-03-13 08:21:52 (EDT)			
d / d	./	2019-03-13 08:24:34 (EDT)	2019-03-13 08:21:52 (EDT)			
d / d	.Trash-0/	2019-03-13 08:24:34 (EDT)	2019-03-13 08:24:34 (EDT)			
✓ r / -	windows.jpg	0000-00-00 00:00:00 (UTC)	0000-00-00 00:00:00 (UTC)			

The 'File Browsing Mode' section below the list states: 'In this mode, you can view file and directory contents. File contents will be shown in this window. More file details can be found using the Metadata link at the end of the list (on the right). You can also sort the files using the column headers.'

16. Still in the upper-right pane, scroll to the right to get the *Meta Data* number for **.Trash-0**. Make a note of this number as you will be using it in the next step.

2019-03-13 08:24:34 (EDT)	2019-03-13 08:24:34 (EDT)	240	48	0	65-144-2
0000-00-00 00:00:00 (UTC)	0000-00-00 00:00:00 (UTC)	0	0	0	0

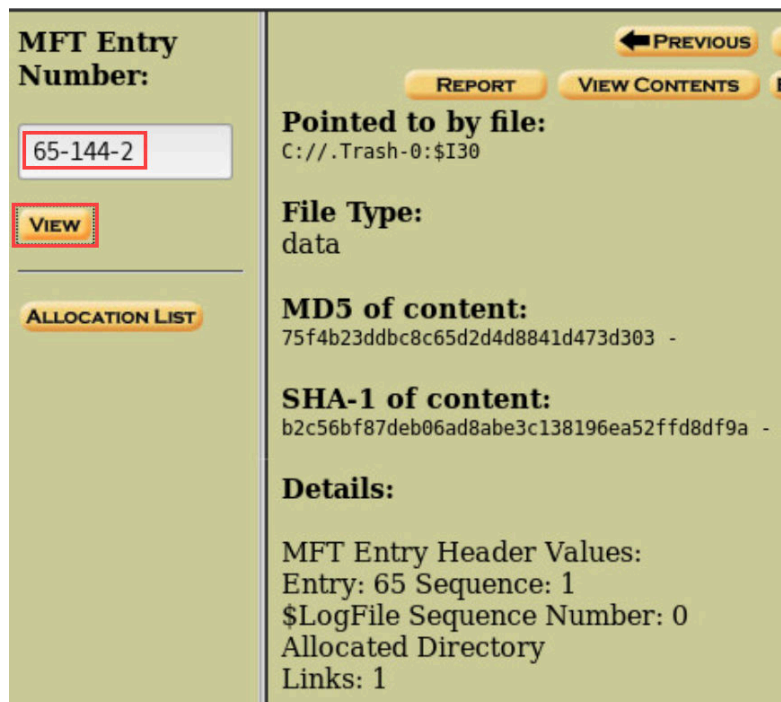
17. Click on **Meta Data** at the top of the page.



The screenshot shows the File Analysis interface with the 'META DATA' tab selected. The 'Directory Seek' pane on the left shows the file details for the selected file:

Permissions	File Name	Size	Modified	MD5	SHA1	SHA256
)		131072	2019-03-13 08:21:52 (EDT)	0	0	10-128-1

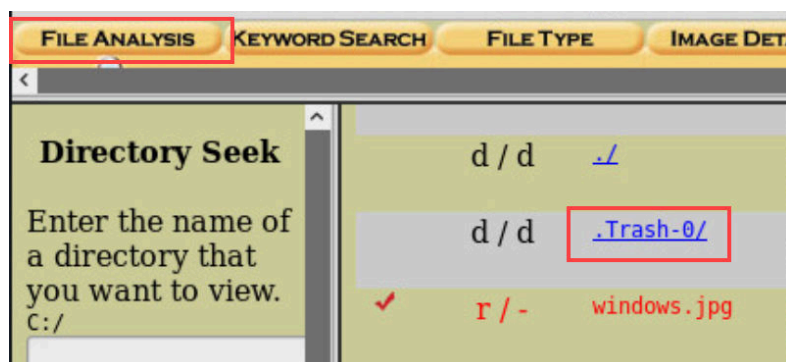
18. Type the previously recorded *Meta Data* number in the left pane and click the **View** button. Examine the details of the directory.



The screenshot shows a web interface for viewing MFT entry details. On the left, there is a section titled "MFT Entry Number:" with a text input field containing "65-144-2" and a "VIEW" button below it. Below the input field is an "ALLOCATION LIST" button. On the right, there are buttons for "PREVIOUS", "REPORT", and "VIEW CONTENTS". The main content area displays the following information:

- Pointed to by file:** C://.Trash-0:\$I30
- File Type:** data
- MD5 of content:** 75f4b23ddbc8c65d2d4d8841d473d303 -
- SHA-1 of content:** b2c56bf87deb06ad8abe3c138196ea52ffd8df9a -
- Details:**
 - MFT Entry Header Values:
 - Entry: 65 Sequence: 1
 - \$LogFile Sequence Number: 0
 - Allocated Directory
 - Links: 1

19. Click on **File Analysis** at the top of the page, and then scroll down in the upper-right pane and click on the **.Trash-0/** directory.



The screenshot shows the "FILE ANALYSIS" tab selected at the top of the page. Below the tabs, there is a "Directory Seek" section with a text input field and a "VIEW" button. The main content area displays a directory tree structure:

- d / d . /
- d / d **.Trash-0/**
- ✓ r / - windows.jpg

20. In the **.Trash-0/** directory, you will find a **files/** folder. Click on it to open.

Current Directory: [C:/](#) / .Trash-0/

[ADD NOTE](#) [GENERATE MD5 LIST OF FILES](#)

DEL	Type	NAME	WRITTEN	Ac
	dir / in	../	2019-03-13 08:24:34 (EDT)	20
	dir / in	./	2019-03-13 08:24:34 (EDT)	20
	dir / in	files/	2019-03-13 08:24:34 (EDT)	20
	dir / in	info/	2019-03-13 08:24:34 (EDT)	20

21. In the **files/** directory, you see that the deleted **windows.jpg** file exists, which was unrecoverable from the captured image. Click on the **windows.jpg** file, and you will see a preview of the file in the lower-right pane.


r / r	windows.jpg	2019-03-13 08:24:13 (EDT)	2019-03-13 08:24:25 (EDT)
-------	-----------------------------	------------------------------	------------------------------

< [Progress Bar]

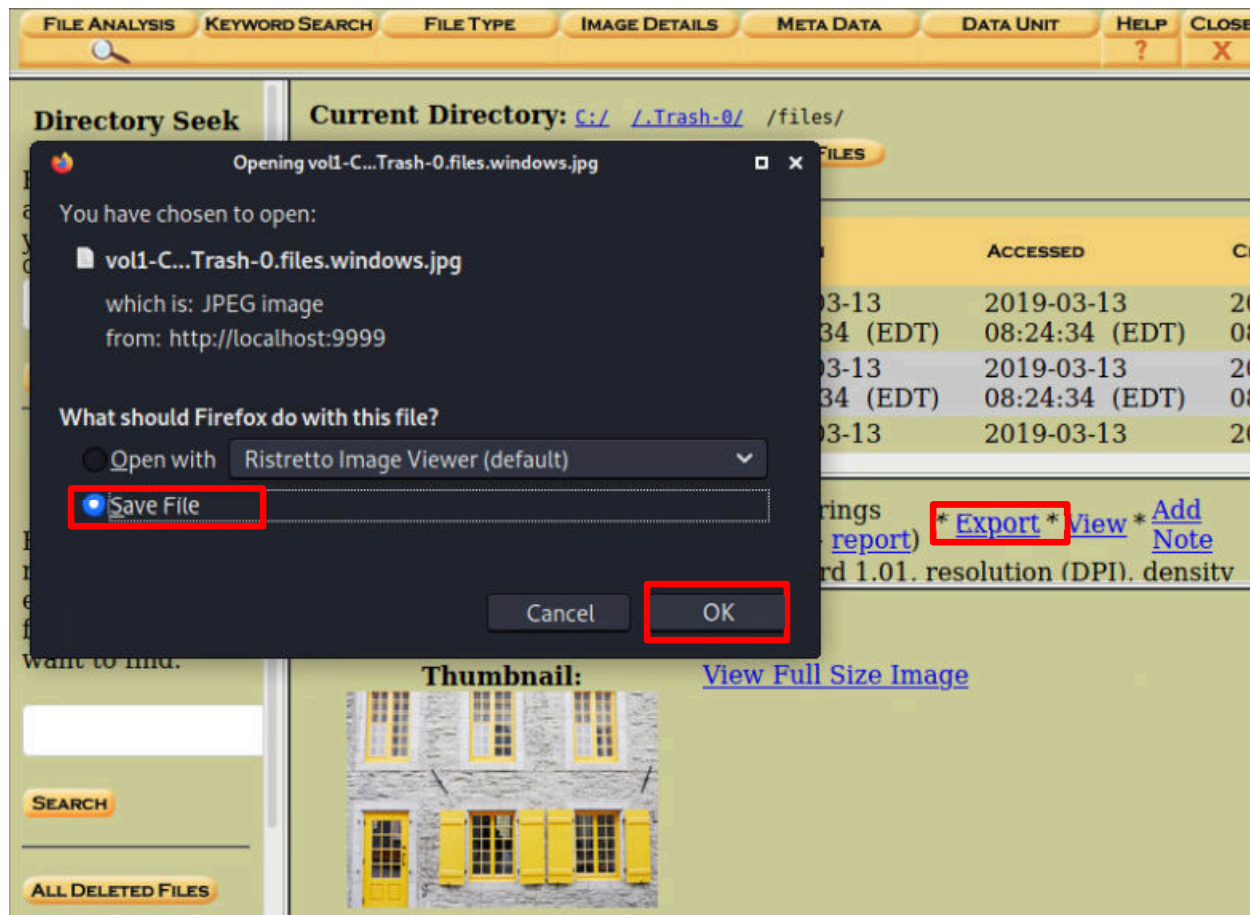
ASCII (display - report) Hex (display - report) ASCII Strings (display - report) * Export *

C:/ .Trash-0/files/windows.jpg

Thumbnail: [View Full Size Image](#)



22. In the lower section, on the right side, click on **Export**. When the popup screen asks, *What should Firefox do with this file?* Click the **Save File** radio button and then click **OK**.



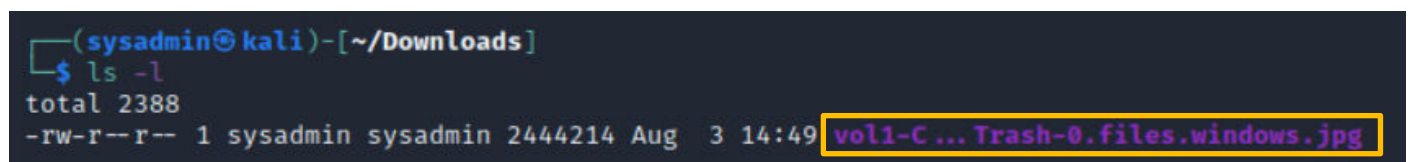
23. Close **Firefox**. Return to the terminal session and stop *Autopsy* by pressing **Ctrl+C**.
 24. In the terminal session, navigate to the **Downloads** directory with the following command:

```
cd Downloads
```



25. View the contents of the **Downloads** directory by issuing the following command.

```
ls -l
```



26. Note that the file you just downloaded is contained in this folder. Move and rename the file to consolidate it with the rest of your files by executing the following command:

```
mv vol1-C...Trash-0.files.windows.jpg  
/home/sysadmin/Desktop/LabFiles/Sleuthkit/windows.jpg
```

```
(sysadmin@kali)-[~/Downloads]  
$ mv vol1-C...Trash-0.files.windows.jpg /home/sysadmin/Desktop/LabFiles/Sleuthkit/windows.jpg
```

27. Navigate to the **Sleuthkit** directory with the following command:

```
cd /home/sysadmin/Desktop/LabFiles/Sleuthkit
```

```
(sysadmin@kali)-[~/Downloads]  
$ cd /home/sysadmin/Desktop/LabFiles/Sleuthkit
```

28. View the contents of the directory with the **ls** command. Note that the **windows.jpg** file is now in your present working directory.

```
ls
```

```
(sysadmin@kali)-[~/Desktop/LabFiles/Sleuthkit]  
$ ls  
autopsy.dd  open.jpg  recovery  sleuthkit.dd  windows.jpg
```

29. To view the **windows.jpg** file, type the following command.

```
ristretto windows.jpg
```

```
(sysadmin@kali)-[~/LabFiles/Sleuthkit]  
$ ristretto windows.jpg
```

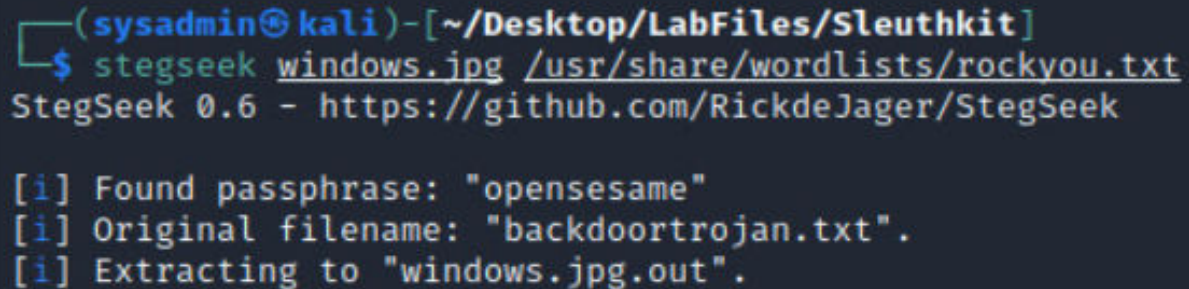


You will see the image that was previewed in **Autopsy**. You, as the security analyst, suspect that this is no ordinary image and that it may contain a hidden file placed inside of it using steganography.

30. Close the image viewer by clicking the **X** in the upper-right corner.

31. There is a tool called *Stegseek* that can quickly scan a picture file looking for embedded files that were inserted using steganography. Type the following command to examine the `windows.jpg` file using a popular wordlist file, `rockyou.txt`.

```
stegseek windows.jpg /usr/share/wordlists/rockyou.txt
```



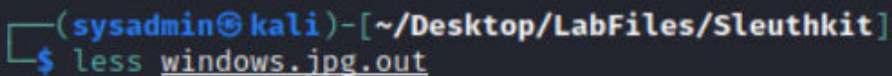
```
(sysadmin@kali)-[~/Desktop/LabFiles/Sleuthkit]
$ stegseek windows.jpg /usr/share/wordlists/rockyou.txt
StegSeek 0.6 - https://github.com/RickdeJager/StegSeek

[i] Found passphrase: "opensesame"
[i] Original filename: "backdoortrojan.txt".
[i] Extracting to "windows.jpg.out".
```

The hidden file has been discovered by *Stegseek*. The file is called **backdoortrojan.txt** and its access passphrase is **opensesame**. The file was extracted to **windows.jpg.out**.

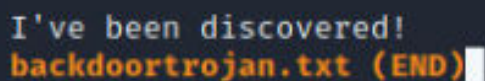
32. View the **windows.jpg.out** file using the following command.

```
less windows.jpg.out
```



```
(sysadmin@kali)-[~/Desktop/LabFiles/Sleuthkit]
$ less windows.jpg.out
```

Congratulations, you have successfully found the source of the virus payload.



```
I've been discovered!
backdoortrojan.txt (END)
```

33. Press `q` to exit the text display and then close the terminal window.
34. This concludes the lab. You may now end the reservation.