Forensics Analysis of the Lustre File System

GRADUATE PROJECT REPORT

Submitted to the Faculty of
The Department of Computing Sciences
Texas A&M University-Corpus Christi
Corpus Christi, TX

In Partial Fulfillment of the Requirements for the Degree of
Master of Science in Computer Science

By

Sowmya Mittakanti
Spring 2011

Committee Members

Dr. Mario Garcia
Committee Chairperson

Dr. John Fernandez
Committee Member

Dr. Hongyu Guo
Committee Member
ABSTRACT

Lustre is a GNU General Public licensed, open-source distributed parallel file system designed, developed and maintained by Sun Microsystems Inc. [7]. The Lustre file system is scalable, secure, robust, and a highly available cluster file system that addresses file I/O needs, such as low latency and extreme performance of large computing clusters. Lustre presents a POSIX interface to its clients with parallel access capabilities to the shared file objects. The Lustre File system is an object-storage model [7]. It provides several abstractions designed to improve both performance and scalability. Lustre technology treats files as objects that are located through Metadata Servers (MDS). File data is stored on the objects of the OSSs[2][4]. The MDS directs actual file I/O requests from Lustre clients to OSSs, which manage the storage that is physically, located on underlying storage devices [1]. Once the MDS identifies the storage location of a file, the entire subsequent file I/O is performed directly between the client and the OSSs. This project involves forensic examination of Lustre File System. This project plan includes four steps: Installation of Lustre File System, Configuration of Lustre File System, Installation of Forensic Tools design and implementation of test cases.
# Table of Content

1. **INTRODUCTION** ........................................................................................................................................... 1

2. **NARRATIVE** ................................................................................................................................................. 4
   2.1 Installing the Lustre Software: ......................................................................................................................... 4
       2.1.1 Installing the Linux operating system: ........................................................................................................... 4
       2.1.2 Creating the virtual volumes: ....................................................................................................................... 5
       2.1.3 Installing the Lustre software: ..................................................................................................................... 6
   2.2 Install the Forensics Tools ............................................................................................................................... 10

3. **EXPERIMENTATION AND RESULTS** ......................................................................................................... 12
   3.1 Case Study 01: Analysis of the File System ................................................................................................. 12
       3.1.1 The Sleuth Kit (TSK) and Autopsy forensics browser: ............................................................................. 12
       3.1.2 Autopsy Forensic Browser: ....................................................................................................................... 16
   3.2 Case Study 02: Recovering of the Deleted Files ............................................................................................ 28
   3.3 Case Study 03: TimeLines of the File Activity .............................................................................................. 33
   3.4 Case Study 04: Analysis on Metadata Target (MDT) of Lustre File System ............................................. 37

4. **CONCLUSION AND FUTURE WORK** ....................................................................................................... 44

5. References ......................................................................................................................................................... 46

Appendix A ............................................................................................................................................................. 47
List of Figures

Figure 1 Lustre File System High-level Architecture: ................................................................. 2
Figure 2 Autopsy main menu: ........................................................................................................ 14
Figure 3 Create a new case: ............................................................................................................ 15
Figure 4 Add a new host: ................................................................................................................ 15
Figure 5 Case and host information: .............................................................................................. 16
Figure 6 Add a new image: .............................................................................................................. 17
Figure 7 Image file details: .............................................................................................................. 17
Figure 8 File analysis mode: .......................................................................................................... 18
Figure 9 Deleted files and their reallocated space: ......................................................................... 19
Figure 10 Directory Seek: ................................................................................................................. 19
Figure 11 Metadata mode: ............................................................................................................... 20
Figure 12 Details of an inode number: ........................................................................................... 20
Figure 13 Details of an inode number: ........................................................................................... 21
Figure 14 Data unit mode: ............................................................................................................... 21
Figure 15 Data of a fragment number in ASCII mode: ................................................................ 22
Figure 16 Data of a fragment number in HEX mode: .................................................................... 22
Figure 17 Data of a fragment number in ASCII String mode: Error! Bookmark not defined. 22
Figure 18 General file details: ....................................................................................................... 24
Figure 19 Sort files by type: ......................................................................................................... 24
Figure 20 Results summary of sort files by type: .......................................................................... 25
Figure 21 Keyword search: ............................................................................................................ 26
Figure 22 File analysis – File browser mode: ................................................................................. 27
Figure 23 Expanded directories: ..................................................................................................... 27
1. INTRODUCTION

This project aims to install the basic Lustre software configuration, which can be used for demonstrating Lustre File Systems. It also installs the forensics tools Sleuth Kit and Autopsy forensics browser for examining the file system. For examining the file system we implement few case studies to use these forensics tools. The goals of this project are:

- Install the Lustre Software.
- Configuration of Lustre File System.
- Install the Forensics tools.
- Implementing Case Studies.

Lustre is a GNU General Public licensed, open-source distributed parallel file system designed, developed and maintained by Sun Microsystems Inc. [7]. The Lustre file system is scalable, secure, robust, and a highly available cluster file system that addresses file I/O needs, such as low latency and extreme performance of large computing clusters. Lustre presents a POSIX interface to its clients with parallel access capabilities to the shared file objects. The Lustre File System is a software only architecture that allows a number of different hardware implementations [3]. The main components of Lustre architecture are the Lustre file system clients (Lustre clients), Metadata Servers (MDS), and Object Storage Servers (OSS). Metadata Servers and Object Storage Servers implement the Lustre File System and communicate with Lustre.
clients. Lustre clients access the Lustre File System via a dedicated network, such as InfiniBand, Ethernet, or other network connections [6].

The Lustre File system is an object-storage model [7]. It provides several abstractions designed to improve both performance and scalability. Lustre technology treats files as objects that are located through Metadata Servers (MDS). File data is stored on the objects on the OSSs. The MDS directs actual file I/O requests from Lustre clients to OSSs, which manage the storage that is physically, located on underlying storage devices [1]. Once the MDS identifies the storage location of a file, the entire subsequent file I/O is performed directly between the client and the OSSs.

Figure 1: Lustre File System High-level Architecture.
Figure 1 gives Lustre File System architecture at higher level. Lustre technology can support a variety of configuration options including a choice of interconnects, single and dual Metadata Servers, and different storage attachment methods for the Object Storage Servers.
2. NARRATIVE

2.1 Installing the Lustre Software

Lustre installation involves two procedures: the preliminary setup and installation of the Lustre software, either from RPMs (Red hat Package Manager) or from source code [3]. Preliminary setup involves the following steps:

- Installing the Linux operating system on all servers and clients.
- Creating the virtual volumes.
- Installing the Lustre software.

2.1.1 Installing the Linux operating system

While any mainstream Linux OS can be used. CentOS (Community Enterprise Operating System) 5.5 were used in the project since it has essentially the same binary as Red Hat Enterprise Linux 5.5. Installing the Linux operating system is straightforward. As the processing system has only one disk, it needed to be partitioned to accommodate both the boot partition and the metadata target(MDT) and object store target(OST) partition.

The disk was split into three partitions:

- Boot partition - 20 GB (enough to house the OS and some applications)
- Swap partition – 2 GB
- Logical volume partition – 50 GB (using a large LVM partition is critical to permit the creation of multiple virtual volumes used for the MDT/OST).

Once the installation was completed, all internal firewalls were disabled to avoid potential problems. All the communication from client to Lustre servers is done via SSH.
2.1.2 Creating the virtual volumes

Virtual volumes need to be created on both Object Store Servers and on the Metadata Server. The Metadata Server requires only one virtual volume and created 5 virtual volumes for each Object Storage Server in this project. Total size of the file system is the sum of the all OSTs of all OSSs. To create virtual volumes first listing the partitions of the system is needed.

`fdisk -l`

This command gives the list of partition disks. Select the logical volume manager partition and create the physical volume on that partition using the command.

`pvcreate <physical device name>`

Next create a volume group:

`vgcreate <vgname> <physical device name>`

Once the volume group is created, create the number of Object Storage Targets (OSTs) in the corresponding volume group. In our project, we have created 5 OSTs on each OSSs and 1 MDT on MDS.
Login into the root and repeat the whole process for creating the OSTs on the second OSS. For MDS, only one virtual volume is needed. So only one MDT is created.

2.1.3 Installing the Lustre software

Before installing and downloading the Lustre software RPMs, connectivity was tested by confirming that ping worked between all systems. The Lustre packages were downloaded free of charge from the following location:

http://www.sun.com/software/products/lustre/get.jsp

The required Lustre packages are:

- lustre-<version>
- kernel-lustre-<version>
- lustre-ldiskfs-<version>
- lustre-modules-<version>
- e2fsprogs-<version>
- lustre-client-<version>
- lustre-client-modules-<version>

Once the required packages were downloaded, Lustre packages were installed. Some Lustre packages are installed on servers (MDS and OSSs), and others are installed on Lustre clients [3]. Lustre packages must be installed in a specific order. Order of the Lustre packages installation.

Lustre kernel RPMs
kernel-lustre-<version>

*Lustre module RPMs*

lustre-modules-<version>

lustre-client-modules-<version> (to be installed on client machines only)

*Lustre utilities RPMs*

lustre-<version>

lustre-ldiskfs-<version>

e2fsprogs-<version>

lustre-client-<version> (to be installed on client machines only)

Install the Lustre packages using the command

```
rpm -ivh <package name>
```

Some of the Lustre RPMs are already installed with the Linux distribution. These RPMs are needed to be updated using the command

```
rpm -Uvh <package name>
```

The installation would yield a number of warnings, primarily due to the lack of the InfiniBand Components. As InfiniBand components may not be needed for Ethernet-based Lustre file systems, these warnings could be ignored.

Once all the RPMs are installed. Reboot the system. When the system is backup, login as root and check the currently running kernel version using the command:

```
uname -a
```

Change the default booting kernel in the `/boot/grub/grub.config` file to `default=0` using the command:
cat /boot/grub/grub.config

After modifying grub.config file, reboot the system. When the system is backup, log in as root and check the currently running kernel version. Now lustre-kernel version will be running.

Repeat this procedure for the MDS, OSSs and client systems. Once all the machines are installed with the Lustre package RPMs, the systems are ready to have the Lustre file system configured on it.

### 2.1.3.1 Configuration of Lustre File System

A Lustre File System consists of four types of subsystems – a Management Server (MGS), a Metadata Target (MDT), Object Storage Targets (OSTs) and clients. The Lustre file system configuration steps should always be performed in the following order:

- Metadata server
- Object Store Servers
- Client

It is possible to set up the Lustre system with many different configurations by using the administrative utilities provided by Lustre [3]. In this project, a simple and standard Lustre file system configuration is build. The configuration also initializes the file system and makes the file system usable once the client has been configured.

Define the module options for Lustre networking(LNET), by adding this line to the `/etc/modprobe.conf` file

```
options lnet networks=<network interfaces that LNET can use>
```

We used tcp0 interface. Therefore it will be options lnet networks=tcp0.
2.1.3.2 Metadata Server:

The following steps describe the Lustre file system configuration on the Metadata Server (MDS).

1. Create a combined MGS/MDT file system on the block device. On the MDS node, run:

   ```bash
   mkfs.lustre --fsname=<fsname> --mgs --mgst <block device name>
   ```

   The default file system name (fsname) is `lustre`.

2. Create a mount point

   ```bash
   mkdir <dir>
   ```

3. Start the MDS node:

   ```bash
   mount -t lustre <block device name> <mount point>
   ```

This completes the Lustre file system configuration on the MDS.

Object Storage Servers:

The following steps describe the Lustre file system configuration on the Object Storage Servers (OSSs):

1. Create mount point for OSTs. In our project we used 5 OSTs names OST1 through OST5:

   ```bash
   mkdir <dir1>
   mkdir <dir5>
   ```

2. Create the OSTs. On the OSS node, run:

   ```bash
   mkfs.lustre --fsname=<fsname> --ost --mgsnode=<Network ID> <mount point>
   ```

3. Start the OSS by mounting the OSTs to the corresponding mount point.

   ```bash
   mount -t lustre <block device name> <mount point>
   ```
4. Repeat this process for each OST.

5. Repeat this process for second OSS.

**Client:**

Setting up the client is very straightforward. All clients mount the same file system identified by the MDS. Use the following commands, specifying the IP address of the MDS server:

```
mkdir <dir>  (creating the mount point)
mount -t lustre <MGS node>:/<fsname> <mount point>
```

Once the mount has been completed, the Lustre file system is ready to use. Verify that the file system started and is working correctly by running the `df`, `dd` and `ls` commands on the client node. For example:

- `lfs df -h`
- `lfs df -ih`
- `dd if=<input file> of=<outputfile>`
- `ls -lsah`

**2.2 Install the Forensics Tools**

The forensics tools for analyzing the file system used in our project are The Sleuth Kit (TSK) and Autopsy forensic browser. Installing the forensics tools is simple and straightforward. Download the latest version of TSK and Autopsy. We used TSK 3.1.3 and Autopsy 2.24. Download the tar.gz files of TSK and Autopsy and extract them [11]. Installation procedure

First install TSK, run the following commands from the TSK directory step by step:

```
./configure
make
make install
```
All the commands must be executed without any errors. Next install the Autopsy forensic browser, run the following commands from the autopsy directory:

`./configure`

Enter the directory that you want use for the evidence locker, in which all the case studies or stored.

`./autopsy`

The output of the command provides an URL, copy and paste the URL in the browser and uses the autopsy forensic browser. Don’t close the terminal when you are working on the autopsy browser, if you close the terminal browser doesn’t work.
3. EXPERIMENTATION AND RESULTS

3.1 Case Study 01: Analysis of the File System

This case study gives the introduction to The Sleuth Kit (TSK) and Autopsy Forensics Browser and execution of the each and every command of the TSK on the Autopsy.

3.1.1 The Sleuth Kit (TSK) and Autopsy forensics browser

The Sleuth Kit is an open source forensic toolkit for analyzing Microsoft and UNIX file systems and disk images [6]. The Sleuth Kit enables investigators to identify and recover evidence from images acquired during incident response or from live systems. The Sleuth Kit is based on two of the most popular and well known open source forensics tools, namely The Coroners Toolkit (TCT) [7] by Dan Farmer and Wietse Venema, and TCTUtils by Brain Carrier. The Autopsy Forensic Browser was initially developed as a graphical interface to TCT, and TCTUtils.

The Sleuth Kit is a collection of UNIX-based command line file system and media management forensic analysis tools. The file system tools allow an examiner to perform a non-intrusive examination of file systems on a suspect computer. The media management tools allow an examination of the layout of disk and other media.

The Sleuth Kit tools will analyze disk of file system images generated by ‘dd’, or similar applications that create a raw image. The tools are designed to be low-level and each performing an individual task, which when used together, can be sued to perform a full analysis.

The Sleuth Kit command lines are divided into layers. The layers are:
File System layer: These conduct the general processing of data from file systems, including layout and allocation structures. This layer includes the following tools.

- **fstat**: Shows details of the file system and statistics, including layout, size and labels.

Content layer: The content layer of a file system contains the actual file content, or data. Data is stored in large chunks, with names such as blocks, fragments, and clusters. Tools within this section process content or data and include the following:

- **dcat**: Extracts the content of a file unit.
- **dstat**: Displays statistics about a given unit in an easy to read format.
- **dls**: Displays the contents of all unallocated units of a file system.
- **dcalc**: Calculates where the data of an image of non-allocated spaces (generated from a DLL) are in the original image [10]. This is used when evidence is found in unallocated space.

Metadata layer: This layer describes a file or directory and contains descriptive data such as dates and sizes as well as the address of data units. Tools within this section process the metadata structures and include the following:

- **ils**: Lists some values of the metadata structures and its content.
- **istat**: Displays metadata information in an ASCII format about a specific structure. This tool will also display the destination of symbolic links.
- **icat**: Displays the contents of the data units allocates to the metadata structure.
- **ifind**: Identifies which metadata structure has allocates a given content unit or file name.
**Human Interface layer:** This layer allows one to interact with files in a manner that is more convenient than directly with the metadata layer [9]. This layer includes the following tools.

- **fis**: Lists file and directory names. This tool will display the names of deleted files as well.
- **ffind**: Identifies the name of the file or directory that has been allocated a given metadata structure.

**Media Management Tools:** These tools take a disk (or other media) image as input and analyses the management structures that organize it.

- **mmls**: Displays the layout of a disk, including the unallocated spaces. The output identifies the type of partition and its length, which makes it easy to use ‘dd’ to extract the partitions. The output is stored based on the starting sector so it is easy to identify gaps in the layout.

**Image File Tools:** This layer contains tools for the image file format. For example, if the image format is a spilt or a compressed image.

- **img_stat**: Displays the details associated with an image file. The output of this tool is image format specific.

**Disk Tools:** These tools can be used to detect and remove a Host Protected Area (HPA) in an ATA disk. A HPA could be used to hide data so that it would not be copied during an acquisition.

- **disk_sreset**: If there is a HPA, then it temporarily removes it so that the full disk can be acquired. When the disk is powered off or reset, then the HPA will exist again.
• **disk_stat**: The important information that is currently given is the actual number of sectors and if there is a Host Protected Area (HPA) on the disk.

**Other Tools:**

• **hfind**: Creates an index of a hash database and perform quick lookups using a binary search algorithm.

• **mactime**: Uses fsl and isl tools output as input to create a timeline of a file activity.

• **sorter**: Analyses a file system to organize the allocated and unallocated files by file type and executes extension checks and hash database checks.

3.1.2 Autopsy Forensic Browser

The Autopsy Forensic Browser is an HTML-Based graphical interface to the command line tools in The Sleuth Kit, showing details about deleted data and file system structures, with results can be accessed using a HTML browser.

Autopsy can create ASCII reports for files and other file system structures. This allows the examiner to quickly make consistent data sheets during the investigation. Audit logs are created on a case; host and investigator level so that actions can easily be recalled.

**Analysis**

The first step of the analysis process is to retrieve the image from the machine, which is to be investigated. The image is extracted using the ‘dd’ command. These tools analysis only dd or raw image files. The following steps do analysis of an image:

**Creation of an Autopsy Case**
Autopsy is setup already and configured on the analysis machine. From the autopsy main menu the ‘New Case’ button needs to be selected in order to start the configuration process for a new case, which is shown in Figure 2.

![Autopsy main menu](image)

Figure 2: Autopsy main menu

New case setup is shown in Figure 3, which needs details like case name, case description, and the details of investigating officers.

![Create A New Case](image)

Figure 3: Create A New Case
After creating the new case setup the computer needed to investigate should be added to autopsy investigating list. Figure 4 shows the process of adding victim computer that is adding new host.

Figure 5 shows the step to add the image files to investigate on host computer. Current page shows a button to add an image file to investigate. Figure 6 shows the step to add the image file on host computer, which shows two options to select the type of the image
on is Disk file and the second is Partition on host computer. To add the image the image must present in the evidence locker. The location of the image file needs to be provided in the location path.

Figure 6: Add A New image
Figure 7: Image File Details

Figure 7 shows the options to select the image file details to investigate. An MD5 hash can be used to verify the integrity of the image.

**File Analysis:**
Figure 8: File Analysis Mode

File analysis mode gives the view of files and directory contents. File content will be shown in the window. More file details can be found using the metadata link at the end of the list. Figure 8 gives the file analysis mode of autopsy tool.

In the file analysis mode using the metadata link that is in File Browsing mode deleted files can be found in red color, which is shown in Figure 9. The space reallocated is also recorded and shown in the figure.
Figure 9: Deleted Files and their Space is Reallocated

<table>
<thead>
<tr>
<th>Directory</th>
<th>Time Written</th>
<th>Time Accessed</th>
<th>Time Changed</th>
<th>Size</th>
<th>UID</th>
<th>GID</th>
<th>Meta</th>
</tr>
</thead>
<tbody>
<tr>
<td>/</td>
<td>2010-10-28</td>
<td></td>
<td>2010-10-28</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>/ftpboot/</td>
<td>2010-10-28</td>
<td>12:57:28 (CDT)</td>
<td>12:57:28 (CDT)</td>
<td>4096</td>
<td>0</td>
<td>0</td>
<td>2542929</td>
</tr>
<tr>
<td>/tmp/</td>
<td>2010-11-30</td>
<td>11:55:17 (CST)</td>
<td>12:13:59 (CST)</td>
<td>4096</td>
<td>0</td>
<td>0</td>
<td>1265313</td>
</tr>
<tr>
<td>/var/</td>
<td>2010-10-28</td>
<td>12:07:26 (CDT)</td>
<td>11:55:17 (CST)</td>
<td>4096</td>
<td>0</td>
<td>0</td>
<td>2123753</td>
</tr>
</tbody>
</table>

Figure 10: Directory Seek

Figure 10 gives the directory structure of the image file on the host computer. Up on clicking the directory name, expands the directory to give the file view.
Metadata mode gives the details about any in the file system. These are the data structures that store the file details. Figure 11 gives the list of nodes in the file system.

![Metadata Mode](image1)

**Figure 11: Metadata Mode**

![Details of an Inode Number](image2)

**Figure 12: Details of an Inode Number**
Figure 13: Details of an Inode Number

Figure 12 and Figure 13 together gives the details of the inode selected. Gives the information the file pointed by the inode, the hashing technique used, group that inode belongs to, file mode, file access time lines, etc.

Figure 14: Data unit mode

Data unit mode shown in Figure 14 gives the content of the Fragments in the file system. And ASCII values of the data present in the fragments are shown in Figure 15. Figure 16
and Figure 17 gives the content of a fragment in HEX mode and ASCII String mode respectively.
Figure 17: Data of a Fragment Number in ASCII Strings mode
Figure 18: General File System Details

Figure 18 shows the file system details in general. It gives the information like file system extension, volume id, number of inodes, inodes in use, free inodes, file access time line, metadata information, content information, and block group information.

In file type mode, Autopsy will examine allocated and unallocated files and sort them into categories and verify the extension. This will allows investigator to find a file based on its type and find “hidden” files. Figure 19 shows the file type sortings.

Figure 19: Sort Files by Type
Results summary page gives the file sorted. Files are categorized into different sections based on the file types. Some of the file types are images, extensions, files skipped and categories. Different file categories are sudo, crypto, data, disk, exec, system, text, etc.

Figure 20 shows the results summary page.

Figure 20: Results Summary of Sort Files by Type
Key word search allows investigator to search for a particular file. Search will be conducted both in allocated and unallocated space and returns ASCII and Unicode formats. Figure 21 shows the search page.

3.2 Case Study 02: Recovering of the Deleted Files

This case study deals with recovering of the deleted files if possible.

Analysis: Creation of an Autopsy Case and search indexes is done same as done in the first case study.

File Analysis:

From the Host Manager, select the Analyze button and then select the File Analysis button, will display the File Browsing mode. The File Browsing mode provides a file manager like view of the contents of the disk image.
Figure 22: File analysis – File Browsing mode

If investigator knows the deleted file, he can directly enter the file name in the directory seek and find the file. If investigator does not know about the deleted file, he can select the All Deleted Files button in the File Analysis mode, which is at left bottom of the screen or press Expand Directories button, which will list all the directories. The file can be found with its path. File browsing mode is shown in Figure 22.

<table>
<thead>
<tr>
<th>Del</th>
<th>Type</th>
<th>Name</th>
<th>Written</th>
<th>Accessed</th>
<th>Changed</th>
<th>Size</th>
<th>UID</th>
<th>GID</th>
<th>Meta</th>
</tr>
</thead>
<tbody>
<tr>
<td>d/d</td>
<td>d</td>
<td>DirMyFiles</td>
<td>00:00:00</td>
<td>00:01:00</td>
<td>00:00:00</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3243737</td>
</tr>
<tr>
<td>d/d</td>
<td>d</td>
<td>DirMyFiles</td>
<td>00:00:00</td>
<td>00:01:00</td>
<td>00:00:00</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3243737</td>
</tr>
<tr>
<td>d/d</td>
<td>d</td>
<td>DirMyFiles</td>
<td>11:56:17</td>
<td>14:55:01</td>
<td>14:56:01</td>
<td>4998</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>d/d</td>
<td>d</td>
<td>DirMyFiles</td>
<td>11:55:17</td>
<td>14:56:01</td>
<td>14:56:01</td>
<td>4998</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
</tbody>
</table>

In this mode, you can view file and directory contents. File contents will be shown in the 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.

Figure 23: Expand Directories
Deleted files are shown in red color. Only the red color files can be recovered. If the file color is dark red it is not possible to recover that deleted file because the deleted file sector is reallocated to the new file. So the data is lost, and cannot be recovered. After finding the deleted file, select the file name from the browser within the File Analysis mode of Autopsy will display the contents of the file. Display of deleted files is shown in Figure 23.

![Figure 23: Deleted Files Display in Autopsy](image)

**Figure 24: Metadata of the Deleted File in ASCII mode**

The metadata for this file can be viewed by selecting the metadata link in the metadata column where the filename is listed. The metadata will detail the sectors where the file was allocated within the disk image, which is shown in Figure 24.

![Figure 24: Metadata of Deleted File](image)
Figure 25: Metadata Analysis

Figure 25 gives the metadata analysis. Metadata information shows that this deleted file was initially allocated sectors are 1052656 - 1052663. Extract these sectors in hope of recovering a copy of the deleted document. There are two ways to extract the information, using the ‘dd’ command or using the export contents button on the top of the screen.

Viewing the deleted file sectors in the data unit mode.

Figure 26, Figure 27 and Figure 28 shows the content of deleted files in ASCII, HEX and ASCII String modes respectively.
Figure 26: Data unit mode of the deleted file in ASCII display

Figure 27: Data unit mode of the deleted file in HEX display
Figure 28: Data unit mode of the deleted file in ASCII String display

Exported content can be viewed through different Hex tools. We need to extract numbers of sectors of the deleted file otherwise some of the information and signatures will be lost and we cannot view the exact information.

### 3.3 Case Study 03: TimeLines of the File Activity

This case study provides the information of the File Activity Timelines of the File System.

**Analysis:** Creation of an Autopsy Case and search indexes was done the same as was done in the first case study.

**File Analysis:**

The log files are binary database that contains the information pertaining to the previous logon sessions including username, port, location and time of last logon.

Autopsy does not provide a direct method for viewing the contents of this file an understandable manner; however the file can be exported to be analyzed folder by an external program. Using the File Activity Timelines, we can view the MAC (Modified Accessed Create) timelines of the files and which user performed activity on the file.
Autopsy provides a mechanism for creating of file activity that is available by selecting ‘File Activity Time Lines’ from the Host Manager.

Figure 29: File Activity Timelines

Figure 29 shows the File Activity Timelines screen. Within the ‘File Activity Timelines’ screen investigator can create a ‘Data File’ that will be used to create a timeline from by selecting the ‘Create Data File’ button.

Figure 30 shows the process of creating data file. The ‘Create Data File’ process allows selecting which partitions to gather meta-data structures from, and also what types of metadata structures are available to include. Metadata structures are available for allocated files, unallocated files and unallocated inodes.

Autopsy has the ability to generate MD5 value for the output files created to ensure the integrity.
Figure 30: Create Data File Results

The data file generated contains the following information for each item of file or directory activity shown in Figure 31:

- File / Directory Name
- Inode
- Access Rights
- Link Count
- User ID
- Group ID
- Size
- A-Time (last Access time of the file)
- M-Time (last Modified time of the file)
- C-Time (file Create time)
- Block Size
Once the data file has been created, Autopsy uses this file to create sorted time lines of events.

The output of the Create Time Line:

![Image](image.png)

**Figure 31: Sample Output of View Timeline**

Figure 31 gives the view of the sample output of TimeLines. The Standard output of the timeline is best viewed in a text editor rather than the Autopsy Forensic Browser. The ‘View Timeline’ mode displays the file system events for a single month on one page at a time.
3.4 Case Study 04: Analysis on Metadata Target (MDT) of Lustre File System

This case study provides the information on the Metadata Target.

In Lustre File System, for each file one inode exists on the Metadata Target. The inode of the MDT does not point to data blocks but instead, points to one or more objects associated with the files. These objects are implemented as files on the OST file systems and contain file date. Each MDT stores file metadata, such as file names, directory structures, and access permissions [8].

Analysis: Extract the Metadata Target image from the Metadata Server using the ‘dd’ command. Creation of Autopsy Case and Search Indexes is done same as in the first case study.

File Analysis:

Image Details mode displays General File System Details of the Metadata Target.
Figure 32 shows the General details about the file system. General file system details provide information of File System Information, Metadata Information, Content Information and Block Information.

For looking into files on the Metadata Target image, select the File Analysis button will display “the file browsing mode”. “File browsing mode” provides a file manager like view of the contents of the MDT image.
Figure 33: File Analysis Mode

Figure 33 shows the File analysis mode in action. In this mode, File manager lists all the directories and files that are present on the MDT. User can expand the directories using the Expand Directories button, which is at bottom left of the screen, and deleted files by using All Deleted Files button.

For analysis of the Lustre file system, we look into the file directories of CONFIGS, OBJECTS, and ROOT. Configs directory contain information about the configuration of all OSTs, MDT, Lustre Clients and mountdata.
Figure 34: Inode report that points to Lustre-Client

Figure 34 gives the inode report. Inode report consists general information about inode as well metadata information.

Figure 35: Configuration Data of the Lustre-OST002

Figure 35 gives the configuration information of the Lustre File System on the host computer.
Autopsy String file can find out of the information of the from which IP address the OSTs and clients are configured. Lustre commands for configuring OSTs. Labels of OSTs.

Objects directory contain information about the objects referenced to each OST shown in Figure 36.

![Figure 36: Files in the object directory](image-url)

Root directory contains all the files that are mounted on to the OSTs shown in Figure 37.

Each file in the root directory contains the metadata information such as file names, directory structures, and access permissions.
Metadata of the file can be viewed by selecting its inode number in the metadata column where the filename is listed shown in Figure 38.

![Figure 37: Files in the Root Directory](image)

Metadata of the file can be viewed by selecting its inode number in the metadata column where the filename is listed shown in Figure 38.

![Figure 38: Metadata of the inode number of its respective file](image)
Autopsy Inode Report

GENERAL INFORMATION

Inode: 426142
Pointed to by file(s):
   /ROOT/lg
   - gd510
   - user
   guide.pdf
MD5 of istat output: db168e9e56edcde4c423da8ee8993d5d3
   - SHA-1 of istat output: 584ca331efcfebaa14ced01862a6d9f641cad9d
   - Image: '/home/evidence/metadataserver/host1/images/metadataserver'
Offset: Full image
File System Type: ext
Date Generated: Wed Dec 15 11:39:34 2010
Investigator: unknown

META DATA INFORMATION

inode: 426142
Allocated
Group: 13
Generation Id: 2518788382
uid / gid: 0 / 0
mode: rwxr-xr-x
size: 0
num of links: 1

Inode Times:
File Modified: Sat Oct 16 02:11:52 2010
Inode Modified: Sat Nov 20 11:29:46 2010

Direct Blocks:

File Type: empty

VERSION INFORMATION

Autopsy Version: 2.24
The Sleuth Kit Version: 3.1.3

Inode report

Inode report provides the metadata information of the file.

Creating File Activity Timelines for the Metadata Target. The output of the File Activity Timelines is
Figure 39: File Activity Timelines of the Metadata Target

File Activity Timelines provides the information of each file mounted on all OSTs shown in Figure 39. It provides the information of file such as, File / Directory Name, Inode, Access Rights, Link Count, User ID, Group ID, Size, Block Size.

4. CONCLUSION AND FUTURE WORK

This project successfully demonstrates setting up the Lustre File system and also describes how this File System functions. The project report depicts the benefits and usefulness of the Lustre File System. Along with the uses Lustre File System is also prone to criminal activities. Therefore this project conducts forensic analysis on the Lustre File System. For forensic analysis on Lustre Files Autopsy Forensic browser was been used in this project. This project also demonstrated the Autopsy Forensic web browser configuration followed by the investigation process. The analysis followed in this project gave the reports of the file system analysis like reports on Files and Directories in the file system, reports on inodes and metadata, content in data fragments.
in three different modes namely ASCII, HEX, and ASCII String modes. Complete summary of the report also provided by Autopsy Forensic Web browser.

This project can be extended in many ways, by implementing some of the Anti-Forensic methods like data hiding, implementation of stenographic techniques on Lustre File System. Thorough research can be performed on recovery of overwritten deleted files on Lustre File System.
5. References


Appendix A

Autopsy string Report

-----------------------------------------------
GENERAL INFORMATION

File: //CONFIGS/lustre-MDT0000
MD5 of file: 2a688c6cdbc49a2be367e45760fb188 -
SHA-1 of file: 4949707724d1056ed873545e745f5f58d37157f -
MD5 of ASCII strings: c642927a6335833844ab48cdad6ea032 -
SHA-1 of ASCII strings: 1e4aa56388390bad2d85d6f99e5544511e7ec1fc -

Image: '/home/evidence/metadataserver/host1/images/metadataserver'
Offset: Full image
File System Type: ext

Date Generated: Wed Dec 15 11:33:44 2010
Investigator: unknown

-----------------------------------------------
META DATA INFORMATION

inode: 393220
Allocated
Group: 12
Generation Id: 2518788187
uid / gid: 0 / 0
mode: rrw-r--r--
size: 21344
num of links: 1

Inode Times:
Accessed: Thu Nov 18 16:45:28 2010
File Modified: Thu Nov 18 16:45:28 2010
Inode Modified: Sat Nov 20 11:39:00 2010

Direct Blocks:
409600 409601 409602 409612 409617 409619

File Type: binary Computer Graphics Metafile

-----------------------------------------------
CONTENT

cconfig_uuid
lustre-mdtlov
lov setup
lustre-mdtlov
lustre-mdtlov_UUID
lustre-mdtlov
lustre-mdtlov_UUID
lustre-mdtlov
lov setup
lustre-MDT0000
add mdt
add mdt
lustre-MDT0000
lustre-mdtlov
lustre-MDT0000
lustre-MDT0000_UUID
lustre-MDT0000
lustre-MDT0000_UUID
lustre-MDT0000
lustre-MDT0000
add mdt
lustre-MDT0000
mdt.group_upcall
lustre-MDT0000
mdt.group_upcall=/usr/sbin/l_getgroups
lustre-MDT0000
lustre.group_upcall
lustre-OST0000
add osc
144.167.113.67@tcp
lustre-OST0000-osc
lustre-mdtlov_UUID
lustre-OST0000-osc
lustre-OST0000_UUID
lustre-OST0000
lustre-OST0000
lustre-OST0000
lustre-OST0000_UUID
lustre-OST0000
add osc
lustre-OST0001
add osc
lustre-OST0001
lustre-mdtlov_UUID
lustre-OST0001-osc
lustre-OST0001_UUID
lustre-OST0001
lustre-OST0001
lustre-OST0001
lustre-OST0001_UUID
lustre-OST0001
add osc
lustre-OST0002
add osc
lustre-OST0002
lustre-mdtlov_UUID
lustre-OST0002-osc
lustre-OST0002_UUID
lustre-OST0002
lustre-OST0002
lustre-OST0002
lustre-OST0002_UUID
lustre-OST0002
add osc
lustre-OST0003
add osc
lustre-OST0003
lustre-mdtlov_UUID
lustre-OST0003-osc
lustre-OST0003-osc
lustre-OST0003
lustre-OST0003_UUID
144.167.113.67@tcp
lustre-mdtlo
lustre-OST0003_UUID
lustre-OST0003
add osc
lustre-OST0004
add osc
144.167.113.67@tcp
lustre-OST0004-osc
lustre-mdtlo_UUID
lustre-OST0004-osc
lustre-OST0004_UUID
144.167.113.67@tcp
lustre-mdtlo
lustre-OST0004_UUID
lustre-OST0004
add osc
lustre-OST0005
add osc
144.167.115.29@tcp
lustre-OST0005-osc
lustre-mdtlo_UUID
lustre-OST0005-osc
lustre-OST0005_UUID
144.167.115.29@tcp
lustre-mdtlo
lustre-OST0005_UUID
lustre-OST0005
add osc
lustre-OST0006
add osc
144.167.115.29@tcp
lustre-OST0006-osc
lustre-mdtlo_UUID
lustre-OST0006-osc
lustre-OST0006_UUID
144.167.115.29@tcp
lustre-mdtlo
lustre-OST0006_UUID
lustre-OST0006
add osc
lustre-OST0007
add osc
144.167.115.29@tcp
lustre-OST0007-osc
lustre-mdtlo_UUID
lustre-OST0007-osc
lustre-OST0007_UUID
144.167.115.29@tcp
lustre-mdtlo
lustre-OST0007_UUID
lustre-OST0007
add osc
lustre-OST0008
add osc
144.167.115.29@tcp
lustre-OST0008-osc
lustre-mdtlov_UUID
lustre-OST0008-osc
lustre-OST0008_UUID
144.167.115.29@tcp
lustre-mdtlov
lustre-OST0008_UUID
lustre-OST0008
add osc
lustre-OST0009
add osc
144.167.115.29@tcp
lustre-OST0009-osc
lustre-mdtlov_UUID
lustre-OST0009-osc
lustre-OST0009_UUID
144.167.115.29@tcp
lustre-mdtlov
lustre-OST0009_UUID
lustre-OST0009
add osc
lustre-mdtlov
lov.stripesize
lustre-mdtlov
lov.stripesize=4194304
lustre-mdtlov
lov.stripesize
lustre-mdtlov
lov.stripecount
lustre-mdtlov
lov.stripecount=0
lustre-mdtlov
lov.stripecount
lustre-mdtlov
lov.stripeoffset
lustre-mdtlov
lov.stripeoffset=65535
lustre-mdtlov
lov.stripeoffset

---------------------------------------------------------------------

VERSION INFORMATION

Autopsy Version: 2.24
The Sleuth Kit Version: 3.1.3
Configuration data of the Metadata Target