

Rochester Institute of Technology

## RIT Digital Institutional Repository

---

Theses

---

2011

### Forensic analysis of VMware hard disks

Manish Hirwani

Follow this and additional works at: <https://repository.rit.edu/theses>

---

#### Recommended Citation

Hirwani, Manish, "Forensic analysis of VMware hard disks" (2011). Thesis. Rochester Institute of Technology. Accessed from

This Thesis is brought to you for free and open access by the RIT Libraries. For more information, please contact [repository@rit.edu](mailto:repository@rit.edu).

Forensic Analysis of VMware Hard Disks

by

Manish Hirwani

Committee Members

Prof. Yin Pan

Prof. Daryl Johnson

Prof. Bill Stackpole

Thesis submitted in partial fulfillment of the requirements for the degree of

**Master of Science in Networking and System Administration**

Rochester Institute of Technology

B. Thomas Golisano College

of

Computing and Information Sciences

05/04/2011

## Acknowledgement

I wish to express my gratitude to each member of my thesis committee without the support and valuable assistance of whom this thesis would not have been possible.

My sincere thanks to Prof. Yin Pan who has been the ever encouraging and motivating force behind my work. She was constantly available for discussions and always gave me prompt advice. Her appreciation of my work has made me work harder each time and has brought forth the best in me at every stage.

I would like to thank Prof. Stackpole for the enthusiasm he has shown in my work throughout my course. His constructive and critical comments extended to me has added his perspective and enriched the contents of my study.

I also thank Prof. Johnson for his constant support and encouragement at every step of the process.

The completion of this dissertation would not have been possible without the valuable assistance of the staff at the NSSA Student Advising Office.

Last but not the least I would like to thank my parents and family for having given me this opportunity to undertake post graduate studies at this renowned institute – RIT - and for their faith in me during my highs and lows throughout these two years.

## **Abstract**

With the advancement in virtualization technology, virtual machines (VMs) are becoming a common and an integral part of datacenters. As the popularity and use of VMs increases, incidents involving them are also on the rise. There is substantial research on using VMs and virtual appliances to aid forensic investigation, but research on collecting evidence from VMs following a forensic procedure is lacking.

This thesis studies a forensically sound way to acquire and analyze VM hard disks. It also discusses the development of a tool which assists in forensic analysis of snapshots of virtual hard disks that are used in VMs. This tool analyzes the changes made to a virtual disk by comparing snapshots created at various stages. Comparing the state of the files in the base snapshot which is believed to be clean with the snapshot which is suspected of being tampered with, forensics investigators are able to identify files that have been recently added, deleted, edited, or modified.

## Table of Contents

Acknowledgement .....	ii
Abstract .....	iii
List of Tables.....	vii
List of Figures.....	viii
1 Introduction.....	1
2 Related Work.....	2
3 Methodology .....	4
3.1 Environment Setup.....	5
3.1.1 Activities performed on the Suspect Machine .....	6
3.1.2 Activities performed on the Analysis Machines.....	7
3.2 Virtual Disk Acquisition .....	8
3.2.1 Guest System Time Skew .....	9
3.3 Forensics Snapshot Analysis Tool.....	9
4 Forensics Snapshot Analysis tool.....	11
4.1 Forensics Snapshot Analysis Menu .....	11
4.1.1 Select Snapshots to Compare.....	12
4.1.2 View Selected Snapshots.....	13
4.1.3 View Files Deleted .....	14
4.1.4 View New Files Added .....	15
4.1.5 View Files Edited [Modification Time] .....	15
4.1.6 View Files Changed [Change Time] .....	15
4.1.7 View SETUID/SETGID Changes .....	16
4.1.8 View Analysis Result Files.....	16
4.1.9 Compute MD5 & SHA1 hashes.....	16

5 Results .....	18
5.1 Image Acquisition .....	18
5.2 Forensics Snapshot Analysis tool .....	19
5.3 Forensics Snapshot Analysis Results .....	20
5.3.1 Files Deleted .....	20
5.3.2 Files Added to Snapshot .....	20
5.3.3 Files Edited .....	21
5.3.4 Files Changed .....	22
5.3.5 Files SETUID/SETGID .....	22
5.4 Miscellaneous Observations .....	23
5.4.1 Editing a File Changes its Inode .....	23
5.4.2 Inode Reallocation .....	23
5.5 Additional Uses for Tool .....	23
6 Limitations & Future Work .....	24
6.1 Possible Methods of Obfuscation .....	24
6.1.1 MAC Times .....	24
6.1.2 Encryption .....	24
6.2 Future Work .....	24
7 References .....	25
Appendix .....	27
MD5 & SHA1 Hashes .....	27
Result Files .....	28
Files Edited .....	28
Files Changed .....	32
Script Source Code .....	37

menu.sh.....	37
select_files.sh .....	40
view_files.sh .....	45
deleted_files.sh .....	47
added_files.sh .....	52
editied_files.sh .....	57
change_files.sh .....	63
setid_changes.sh .....	69
result_files.sh .....	74
check_md5.sh.....	76

## List of Tables

Table 1: List of files that make up a Virtual Machine .....	1
Table 2: Hardware and Software configuration of Host Laptop .....	5
Table 3: Hardware and Software configuration of Thesis Suspect Machine .....	6
Table 4: Hardware and Software configuration of Forensics Analysis Machine.....	6



## List of Figures

Figure 1: Flow Chart of the process followed. ....	4
Figure 2: Environment Setup.....	5
Figure 3: FTK acquiring .vmdk in raw (dd) format.....	8
Figure 4: EnCase acquiring .vmdk in raw (dd) format. ....	9
Figure 5: Main Menu of Forensics Snapshot Analysis.....	11
Figure 6: Snapshot selection procedure for Forensics Snapshot Analysis tool.....	13
Figure 7: The “View Selected Snapshots” of the tool .....	13
Figure 9: Output of the View Files Deleted analysis script.....	15
Figure 10: The View Analysis Result Files menu.....	16
Figure 11: Compute MD5 Hashes menu .....	17
Figure 12: MD5 & SHA1 hashes generated after conversion to raw by FTK.....	18

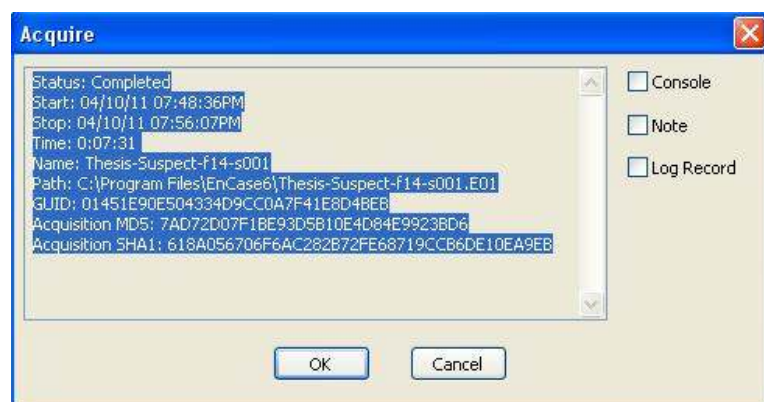


Figure 13: MD5 & SHA1 hashes generated after conversion to raw by EnCase. ....	19
Figure 14: MD5 & SHA1 hashes obtained from the Forensics Analysis tool after analysis. ....	19

# 1 Introduction

Traditionally computer systems such as desktops and servers have been considered physical devices. With the introduction of virtualization in the IT industry, this may no longer be the case. Owing to the benefits of virtualization, virtualization is becoming a widely adopted practice across organizations of various sizes. VMware is a popular provider of virtual machine (VM) software and holds a large share of the market. Products offered by VMware include VMware Workstation, VMware Server, and VMware ESXi among others. With the growing trend in virtualization, more and more production systems, workstations and desktops are being virtualized. Being a popular provider, VMware virtual environments are likely to be encountered by forensic investigators and hence this research will focus on analysis of VMware products.

VMware VMs are implemented using virtual adapters for devices such as network cards, memory, etc. The VM is stored in a set of files. VMware Workstation creates files with extension like .vmx, .vmdk, .vmem, etc. Below is a table with a short description of the files that make up a virtual machine posted on VMware's website under "What Files Make up a Virtual Machine?" [20].

Extension	Description
.vmdk	virtual hard drive for the guest operation system
.vmem	backup of the virtual machine's paging file
.vmsn	snapshot file
.vmsd	snapshot metadata
.nvram	virtual machine bios information
.vmx	virtual machine configuration file
.vmss	virtual machine suspended state file

Table 1: List of files that make up a Virtual Machine

As the popularity and use of VMs increases, greater number of incidents are occurring involving VMs. The conventional methods for incidence response and acquiring evidence, such as pulling the plug of a machine that is powered on and disconnecting the hard disk to gather evidence, may not be appropriate to forensics of VMs. The aim of this research is to analyze existing forensics methods and suggest a methodology to analyze a virtual hard disk created by VMware that is forensically sound.

## 2 Related Work

According to Kruse & Heisere the conventional forensic process can be broadly classified into four main phases, viz. Acquire, Preserve, Analyze and Report [10]. The acquired state of the process involves capturing as much system volatile data as possible, then powering down the system and creating a forensic image of all the storage devices that are found [5]. A forensic image of a device is a bit by bit copy of the drive. The bit stream copy can be either stored as a file on another device or can directly be copied to another drive of similar or greater capacity. The bit stream copy of the storage drive is generally acquired using a dd based tool [15]. This image is stored in a raw format supported by a dd or a propriety format which is typically based on dd [17]. The acquired image is either an identical copy of the storage device or the data from the device which is stored in a format that can be used to evaluate the contents and can be presented in court as evidence. Analysis can be conducted using any of the many open-source or propriety tools available such as Sleuthkit, Forensics ToolKit or EnCase.

Most of the research conducted in the area of virtualization and forensics makes use of VMs as forensics tools. VMs can be used to conduct analysis of evidence. VMs provide the examiner the ability to have a clean operating system without having to wipe a drive and install a fresh operating system on it for every new case. Helix [7] & Penguin Sleuth Kit Virtual Computer Forensics and Security Platform [6], are popular Linux based operating systems tailored for forensics acquisition and analysis, are readily available as virtual appliances that can be used with VMware products. Similarly, other virtual appliances are available which use virtualization to assist in conducting a forensic investigation.

Mrdovic et al. used a tool called Live View, which creates a VMware VM from a raw image of a drive or a physical drive [14]. This enables the investigator to boot up the disk in a virtual environment and gain an interactive, user-level perspective of the suspect's environment. All this is done without modifying the image or the physical drive and is considered to be forensically sound.

As incidents related to VMs are on the rise, they have caught the attention of forensics experts. New methods to collect evidence from VMs are needed. Fiterman and Durick in their article titled "Ghost in the Machine: Forensic Evidence Collection in the Virtual Environment" point out that tools and options that enable an examiner to investigate virtual data are currently limited [8]. They also suggest best practices that can be followed for virtualized environments and point out that they are not very different from those followed for non-virtualized environments. They suggest creating a snapshot of the machine and collecting the associated files such as .vmdk and .vmsn. Similar concerns regarding the absence of forensics tools and procedures for VM analysis are raised and methodologies are proposed by Beek [4]. He also suggests a tool which compares the memory files (.vmem) of snapshots created by VMware products for any new files or processes.

Bares in his research studied the amount of data that could be recovered from NTFS partition which has a VMware Server hosting VMs. He compares the data recovered based on the memory assigned to the virtual machines [2]. He concluded that memory did not have a significant effect on the amount of data that could be recovered. He discovered that the number of files recoverable was inversely proportional to the memory allocated to the VM. His research also showed that lesser amount of data was recoverable if the VM was incorrectly shut down as compared to when it was shutdown gracefully.

In conclusion, there is abundant research on using VMs and virtual appliances to aid forensic investigation, but research on collecting and analyzing evidence from VMs is lacking. Forensics examiners need to understand the virtual environments they might encounter and which are the files of interest that should be acquired. Tools need to be identified that can be used to conduct such analysis in a forensically sound manner [4].

### 3 Methodology

The use of VMs in corporate and personal environments is rapidly increasing, 18% of servers were virtualized in 2009 and that grew to 25% in 2010. It is expected that by 2012, about half of the servers hosted will be virtualized and hosted virtual desktops will reach 49 million units by 2013 [13]. With the growing number of virtual systems it becomes imperative that a methodology to analyze virtual systems is developed. Many systems carry out critical tasks which cannot be stopped. If such systems are compromised, or are suspected of being compromised, they cannot be taken offline for analysis. In such a scenario, it becomes important to conduct a live analysis of the system.

However, when live analysis is carried out the investigator may lose information that is in the memory or network connections that were open may be terminated. According to Kurse & Heiser [10], the common practice to conduct forensic analysis of a physical machine is to take the machine offline at some point in time. The machine hard disk is then imaged, and its data is acquired for analysis.

When VMs are involved in an incident, the VM is usually suspended or a snapshot of the machine is created to preserve the processes and network status for forensics analysis. In this section, the forensics technique used to acquire, preserve, and analyze snapshots of disk images created by the VMware Snapshot utility will be discussed.

Figure 1 shows the basic flowchart of the process followed during the research of this paper and the various components/processes that were carried out to conduct this research.

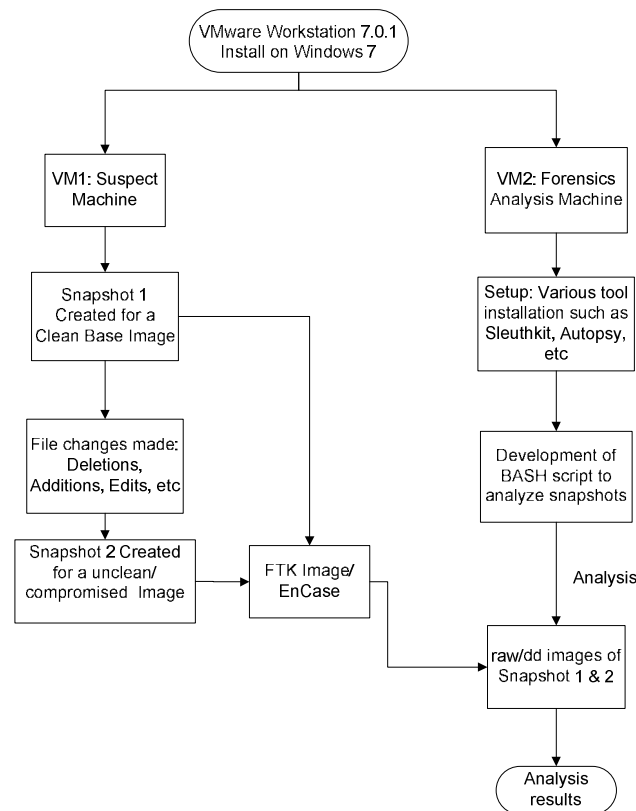


Figure 1: Flow Chart of the process followed.

### 3.1 Environment Setup

The author sets up a virtual environment to test his hypothesis and develop the proposed tool. VMware Workstation 7.0.1 was installed on a Windows 7 Home Premium operating system with a New Technology File System (NTFS) partition. Two Fedora 14 operating systems were installed virtually and stored on a File Allocation Table (FAT) partition. One of the Fedora 14 VMs was used as a suspect machine and the other one as a forensics analysis machine. The setup is depicted in the figure below:

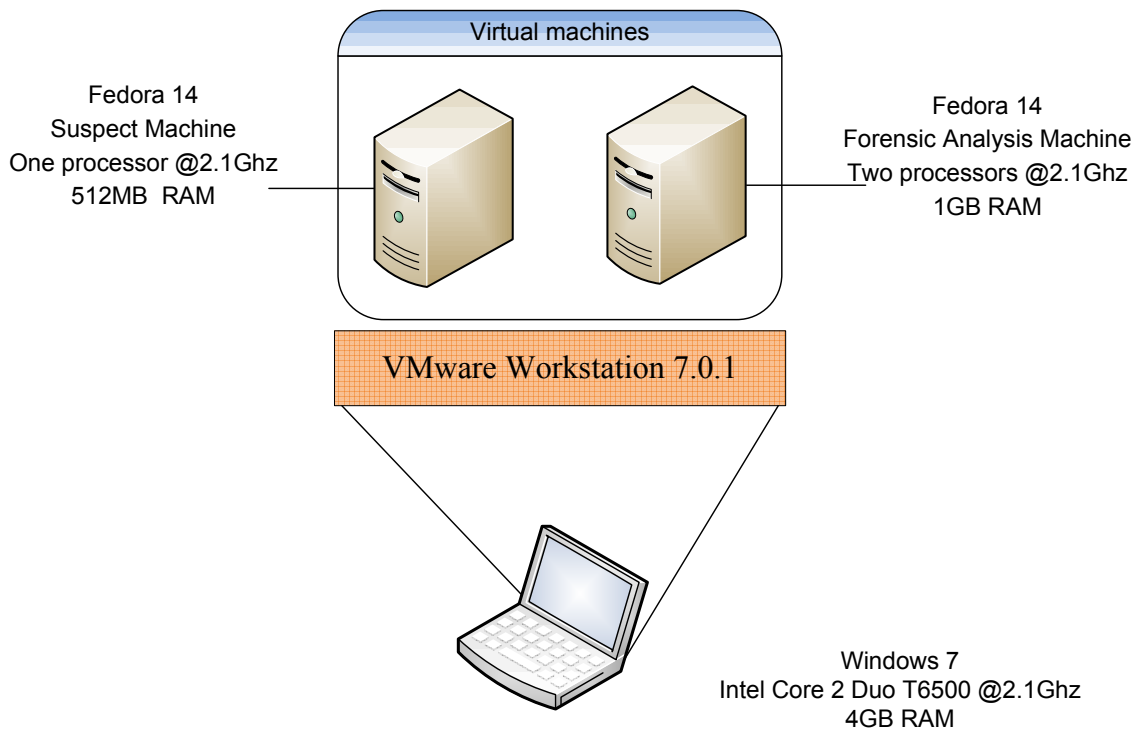


Figure 2: Environment Setup

The configuration of the Window 7 and Fedora 14 systems is tabulated below.

Item	Version
Laptop	HP dv4-1435dx
Processor	Intel Core 2 Duo T6500 @2.10GHz
Memory	4GB
Operating System	Windows 7 Home Premium
Virtualization Platform	VMware Workstation 7.0.1 32 bit

Table 2: Hardware and Software configuration of Host Laptop

Item	Version
Virtual Processors	One @2.10GHz
Memory	512MB
Disk space	10GB
Operating System	Fedora 14

Table 3: Hardware and Software configuration of Thesis Suspect Machine

Item	Version
Virtual Processors	Two @2.10GHz
Memory	1GB
Disk space	40Gb
Operating System	Fedora 14

Table 4: Hardware and Software configuration of Forensics Analysis Machine

Once basic installation was completed, various activities were carried out on the Suspect and Analysis machines to prepare them for the research planned.

### 3.1.1 Activities performed on the Suspect Machine

Once the Fedora 14 operating system was installed on the Suspect Machine the following activities were performed:

- A user account “manish” was created
- Selinux was put into permissive mode
- NMAP was installed using yum
- TrueCrypt was installed using yum
- A file named del\_me.txt was created on the Desktop of the account manish
- A TrueCrypt volume “true\_crypt.txt” was created and one text file was placed inside it.

Once the above operations were performed, a snapshot of the system was created. This snapshot was established as the clean base snapshot and any further snapshots would be compared against it. After the snapshot was created, the following changes were made to the system:

- Selinux was disabled by editing the file `/etc/sysconfig/selinux`
- IPTABLES were flushed at boot by adding the following instructions in the `/etc/rc.local` file
  - `/sbin/iptables -F`
  - `/sbin/iptables -t nat -F`
- The NMAP, TUNE2FS & MODPROBE binaries were setuid
- The SERVICE & RMMOD binaries were setgid
- The files `added.txt` & `del_me2.txt` were created
- The file `del_me.txt` was deleted and `added2.txt` was created
- The file `del_me2.txt` was deleted next
- Files were added and deleted from TrueCrypt volume `true_crypt.txt`

Once these actions were carried out, another snapshot of the system was created and this was labeled as the unclean or compromised snapshot.

### 3.1.2 Activities performed on the Analysis Machines

Next, the Forensics Analysis machine was setup and the following steps were carried out after the installation of Fedora 14 operating system:

- A user account “manish” was created
- Selinux was put into permissive mode
- IPTABLES were flushed at boot by adding the following instructions in the `/etc/rc.local` file
  - `/sbin/iptables -F`
  - `/sbin/iptables -t nat -F`
- VMware tools for Linux were installed.
- Sleuthkit and Autopsy Forensic Browser were installed.

Once the setup for the analysis machine was complete, a BASH script was written which analyzed the snapshots and compared them.



### 3.2 Virtual Disk Acquisition

The aim of forensics is to minimize contamination and ensure admissible evidence. To accomplish this, the digital evidence acquisition process has to follow an appropriate procedure. Acquiring non-volatile data from a hard disk entails many steps [10]. These steps are briefly discussed; a machine is first powered off by disconnecting the power supply from the machine (i.e. pulling the plug). The hard disk is then removed from the suspect machine and connected to a forensic analysis machine. The hard disk is then imaged using any of the many tools available for imaging a disk such as dd, FTK Imager, EnCase, etc. This image is then used by a forensics investigator to conduct analysis.

When working with suspended VMware images, there are two options for acquiring the virtual disks - either resuming the suspended system, then use bit-by-bit copy or to directly work with the VMware files. The problem with resuming a VM is that during the resume process, many files stored on the hard disk are changed, which may destroy evidence. Another disadvantage of resuming the suspended VM is that, there is a loss of information stored in the memory as the state of the VM changes. This information could be vital to the investigation being carried out.

The methods suggested by the author to create a snapshot of the VM and then use VM files that are stored on the host system does not suffer the shortcomings of resuming the suspended VM. Upon taking a snapshot, the state of the hard disk is preserved and any changes to the disk are stored in a separate file. The virtual memory of the VM is stored in a file and any state changes are written to another virtual memory file. The author concludes that, this would be a better method to conduct analysis as the evidence is preserved and can be used as forensically sound and admissible evidence.

Both EnCase and FTK support conversion of .vmdk files to raw (dd) format. When FTK is pointed to a snapshot for converting it to a raw image, it converts the snapshot along with any previous snapshots and the base vmdk files (see Figure 3).

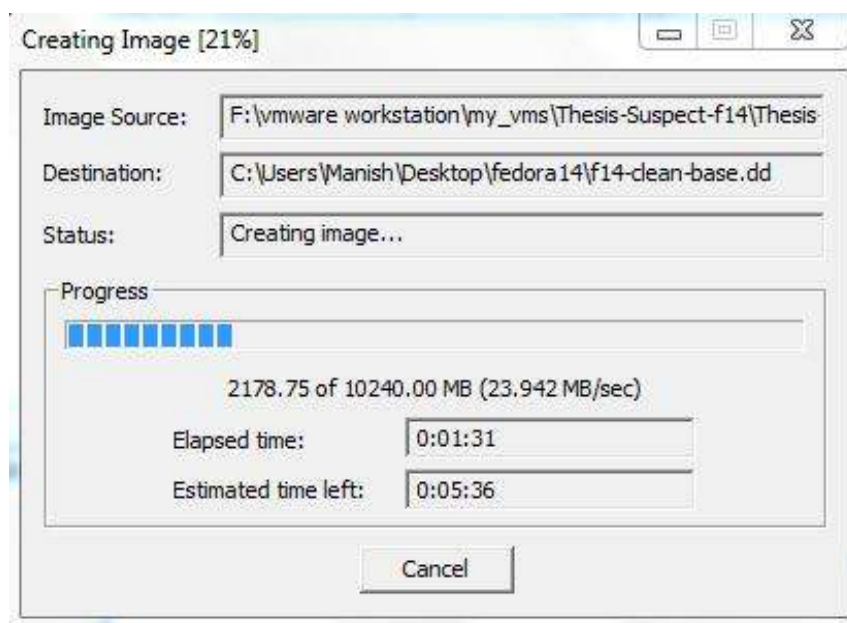


Figure 3: FTK acquiring .vmdk in raw (dd) format.

When EnCase is pointed to the base clean vmdk files, it successfully converts them to raw/dd format, but when EnCase is pointed to a later snapshot, it only converts the delta that is created after snapshotting a VM. If we wish to have EnCase include the previous snapshots and the base images as well, we need to assimilate a flat vmdk file by using utilities that are packaged with VMware Workstation, namely vmware-vdiskmanager.exe. Vmware-vdiskmanager.exe creates one vmdk file which includes the snapshot that it is pointed to, and any previous snapshots and the base vmdk files. This single vmdk file can then be converted to raw/dd using EnCase (see Figure 4).

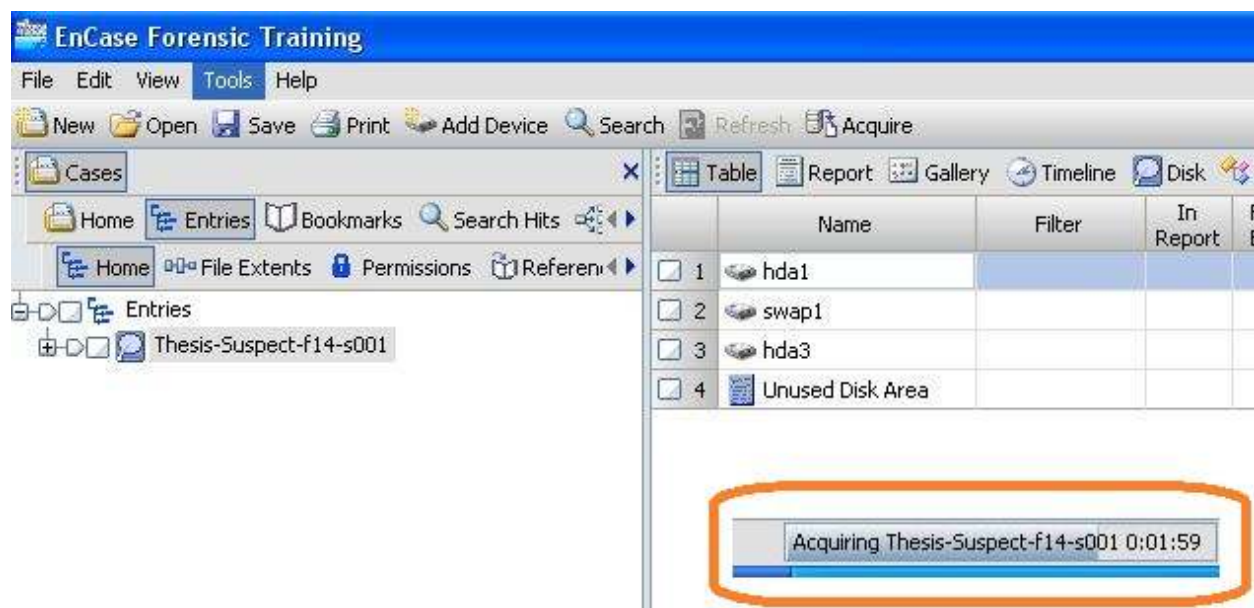


Figure 4: EnCase acquiring .vmdk in raw (dd) format.

Using these techniques, the contents of the virtual hard disk can be acquired even if the VM is powered on, since a snapshot of this machine is created at the time of incident response.

### 3.2.1 Guest System Time Skew

When the guest system is being acquired it is critical for the incident response professional to record the time of the guest operating system as well as the host operating system. The time of the guest operating system could be skewed and if this is recorded the forensic examiner can make more definitive statements about the activities that took place and also accurately pin point when they did.

If the guest system clock is synchronized with the host system clock, the incident response professional should make sure to check if the host system time is correct else he should record the skew. If the guest is using an external source besides the host system clock to synchronise the time, any skew that exists should be recorded. The Forensics Snapshot Tool, as well be seen later, takes into account the time skew of the guest operating system to conduct analysis.

## 3.3 Forensics Snapshot Analysis Tool

With the acquired image from section 3.2, one can use open and commercial forensics tools, for example, EnCase, FTK, and Sleuthkit etc, to conduct a forensics analysis. However, with the additional

files created from the VM and features supported by VMware, are there other efficient techniques that could assist forensics investigation?

VMware offers the Snapshot feature which allows one to freeze the state of a VM at a given point of time [20]. If a VM was suspended and a snapshot was taken at the time of incidence response, the forensics investigator may simply conduct a live analysis, since the state of the machine is preserved in a snapshot, any changes made by the investigator can be reverted to the original state.

In this research the author developed a *Forensics Snapshot Analysis* tool that compares the snapshot with a clean staged snapshot to identify possible malicious activities.

This tool is written in Bash script and incorporates existing tools, such as Sleuthkit, md5sum and sha1sum to verify the integrity of the evidence and also conduct forensic analysis. The script extracts files from both the clean and the compromised image snapshots. Then a comparison is made to determine the changes that have been made such as files created, changed or deleted. The tool is also capable of identifying MAC time changes, content changes and permission changes. These modified files are reported and can then be further investigated by a forensics examiner.

## 4 Forensics Snapshot Analysis tool

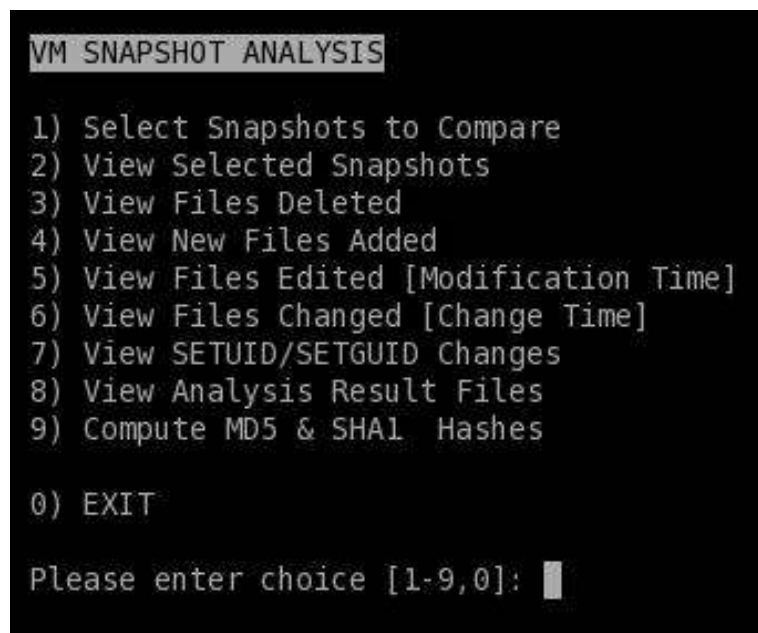
The Forensics Snapshot Analysis tool is written using BASH. It analyzes the changes made to a recent snapshot of a virtual machine by comparing it to an earlier snapshot. When a VM is suspected of being compromised, a snapshot can be taken and this snapshot can be compared to a snapshot that is believed to be clean.

The Forensics Snapshot Analysis tool helps in identifying new files or binaries added to the system, files deleted from the system, files edited, files changed and files which have been modified by setting the SETUID and SETGID bit. The tool achieves this by comparing the list of files found in the base snapshot and the list of files found in the compromised snapshot. It also compares the modification time and changed time of the files found in both snapshots. The tool incorporates Sleuthkit binaries and uses them to extract evidence from the snapshot.

The Forensics Snapshot Analysis tool requires that the two snapshots are of the same VM and the snapshots are converted to raw/dd format. The tool can then successfully analyze the changes that have been made to the recent snapshot by comparing these two snapshots.

### 4.1 Forensics Snapshot Analysis Menu

The Forensics Snapshot Analysis tool is menu driven and can be used by novice professionals as well as experts. Shown in Figure 5 is the Main Menu of the tool.



```
VM SNAPSHOT ANALYSIS

1) Select Snapshots to Compare
2) View Selected Snapshots
3) View Files Deleted
4) View New Files Added
5) View Files Edited [Modification Time]
6) View Files Changed [Change Time]
7) View SETUID/SETGID Changes
8) View Analysis Result Files
9) Compute MD5 & SHA1 Hashes

0) EXIT

Please enter choice [1-9,0]: █
```

Figure 5: Main Menu of Forensics Snapshot Analysis

As can be seen from Figure 5, the script is rich with features and allows the user to select the snapshots they want to analyze, view selected files, and conduct various analyses on the selected snapshots.

The tool allows the user to perform the following actions:

1. Select Snapshots to Compare
2. View Selected Snapshots
3. View Files Deleted
4. View New Files Added
5. View Files Edited [Modification Time]
6. View Files Changed [Change Time]
7. View SETUID/SETGID Changes
8. View Analysis Result Files
9. Compute MD5 & SHA1 Hashes

To use any of the options, the user must select the operation he wants to perform. Once the option is selected, the script will perform the action that it is instructed to follow.

#### **4.1.1 Select Snapshots to Compare**

This option of the tool allows the user to select the snapshots for analysis of evidence.

Figure 6 below shows the procedure to select the snapshot that will be analyzed. As seen from the figure only RAW files can be used with the tool. To select the snapshot the complete path of the snapshot must be input. The snapshots must be stored on locally attached storage device.

The user can also input the time skew of the guest operating system. The Forensic Snapshot Analysis tool will then use this time skew to display the corrected times. The source for this script can be seen in the Appendix and is titled *select\_files.sh*.

```
VM SNAPSHOT ANALYSIS

Select Snapshots to Compare

Note 1: Please input path of RAW files only.

Note 2: Please input complete file path.

Note 3: Please make sure the snapshots belong to the same VM.

Please input path of clean snapshot which
will be used as a baseline for comparison.
PATH 1: /home/manish/data/f14-clean-flat-files/f14-clean-flat-files-dd.001

Please input path of suspected snapshot which
will be against the baseline for comparison.
PATH 2: /home/manish/data/f14-changes-flat-files/f14-changes-flat-files-dd.001

Please input the time skew of the system in seconds, else enter 0:-100

Both files seem to be valid, paths recorded.
Please press <ENTER> to continue.█
```

Figure 6: Snapshot selection procedure for Forensics Snapshot Analysis tool

#### 4.1.2 View Selected Snapshots

Using this option a user can verify the files that have been selected for analysis.

To verify that the correct files have been selected, the tool offers the functionality to view the files that have been selected for analysis. This can be done via the “View Selected Snapshots” option. The figure below shows this functionality. The user can also verify the time skew of the guest system that was input earlier. The source for this script can be seen in the Appendix and is titled *view\_files.sh*.

```
VM SNAPSHOT ANALYSIS

View Select Snapshots

Baseline snapshot: /home/manish/data/f14-clean-flat-files/f14-clean-flat-files-dd.001

Suspected snapshot: /home/manish/data/f14-changes-flat-files/f14-changes-flat-files-dd.001

Time Skew: -100 seconds

Please press <ENTER> to continue.█
```

Figure 7: The “View Selected Snapshots” of the tool

### 4.1.3 View Files Deleted

Using this option, the user can conduct analysis on the selected snapshots to determine which files were deleted by comparing the files present in the clean snapshot to those that are present in the compromised snapshots. Once the option is selected, the script will analyze the snapshot image and will display the image partition table. The user can then select the partition they want to analyze and the script will collect evidence from the partition as can be seen from the figure below. The source for this script can be seen in the Appendix and is titled *deleted\_files.sh*.

```

VM SNAPSHOT ANALYSIS
View Files Deleted

The partition are listed below:

Disk f14-clean-flat-files-dd.001: 10.7 GB, 10737418240 bytes
255 heads, 63 sectors/track, 1305 cylinders, total 20971520 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x0005e2f5

    Device Boot      Start         End      Blocks   Id  System
f14-clean-flat-files-dd.001p1  *           2048        411647       204800   83  Linux
f14-clean-flat-files-dd.001p3          2508800       20971519      9231360   83  Linux

Please enter the start of the partiton sector to analyze: 2508800
Content of clean snapshot are being recorded, please wait:
|                                     |

Content of suspect snapshot are being recorded, please wait:
|                                     |

```

Figure 8: The tool collecting evidence from the partition selected by the user.

As can be seen, the tool also displays a progress bar while collecting the evidence. Once the evidence is collected from both snapshots, the tool analyzes the evidence collected and displays files of interest to the user.

The image below is the output of the script that shows files that were deleted.

```
Following are the files that have been deleted:

/etc/mtab~2122 (deleted)
/etc/systemd/system/graphical.target.wants/firstboot-graphical.service (deleted)
/home/manish/Desktop/.added3.txt.swp (deleted)
/home/manish/Desktop/del_me.txt
/home/manish/.local/share/gnote/73b05277-b147-4cec-ad05-31b957eba200.note
/home/manish/.local/share/gnote/73b05277-b147-4cec-ad05-31b957eba200.note.tmp (deleted-realloc)
/home/manish/.local/share/gvfs-metadata/home-2a4bc7a9.log (deleted)
/home/manish/.local/share/gvfs-metadata/home-759096e1.log
/home/manish/.local/share/gvfs-metadata/home-759096e1.log.ZLIORV (deleted-realloc)
/home/manish/.local/share/gvfs-metadata/home-8RVORV (deleted-realloc)
/home/manish/.mozilla/firefox/5ixj0qol.default/lock (deleted)
/home/manish/.mozilla/firefox/5ixj0qol.default/sessionstore.js (deleted-realloc)
/$OrphanFiles/OrphanFile-174250 (deleted)
/$OrphanFiles/OrphanFile-181966 (deleted)
/root/.recently-used.xbel.VGYLRV (deleted-realloc)
/root/.xauthNrPy2u (deleted)

Please press <ENTER> to continue.█
```

Figure 9: Output of the View Files Deleted analysis script

Along with displaying the files of interest, the script also stores the results for each operation in separate files. Before displaying the results to the user, the script notifies the user with the filename and location the result is stored in. A user can later view the stored results.

#### 4.1.4 View New Files Added

Using this option, the user can conduct analysis on the selected snapshots to determine which files were added to the compromised snapshot by comparing the files present in the clean snapshot to those that are present in the compromised snapshots. The procedure of gathering evidence, analyzing it and reporting is the same as described for the “View Files Deleted” option. The source for this script can be seen in the Appendix and is titled *added\_files.sh*.

#### 4.1.5 View Files Edited [Modification Time]

Using this option, the user can conduct analysis on the selected snapshots to determine which files were edited after the snapshot of the clean system was taken. This is done by comparing the modification time of each file present in the clean snapshot to those that are present in the compromised snapshot. The procedure of gathering evidence, analyzing it and reporting is the same as described for the “View Files Deleted” option. The source for this script can be seen in the Appendix and is titled *edited\_files.sh*.

#### 4.1.6 View Files Changed [Change Time]

Using this option, the user can conduct analysis on the selected snapshots to determine which files were changed (i.e change time or ctime differs) after the snapshot of the clean system was taken. This is done by comparing the change time of each file present in the clean snapshot to those that are



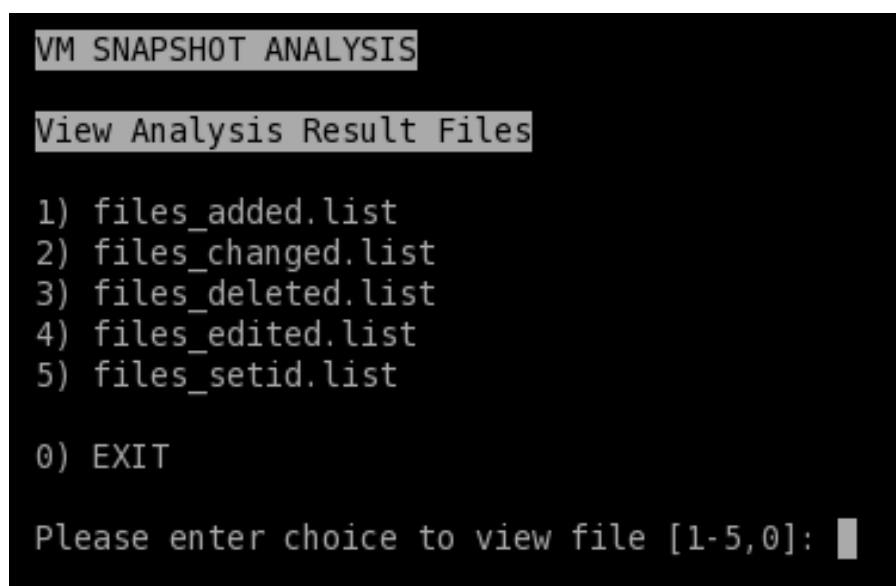
present in the compromised snapshot. The procedure of gathering evidence, analyzing it and reporting is the same as described for the “View Files Deleted” option. The source for this script can be seen in the Appendix and is titled *change\_files.sh*.

#### 4.1.7 View SETUID/SETGID Changes

Using this option, the user can conduct analysis on the selected snapshots to determine which files have SETUID or SETGID permissions changed. This is done by comparing the permissions of files that were changed since the snapshot of the base image was taken and comparing them to the permissions of the files that are present in the clean snapshot. The procedure of gathering evidence, analyzing it and reporting is the same as described for the “View Files Deleted” option. The source for this script can be seen in the Appendix and is titled *setid\_changes.sh*.

#### 4.1.8 View Analysis Result Files

As described previously, the script stores the files of interest found after analysis in result files and reports the file name to the user when analysis is complete. Viewing these files can be done by selecting the “View Analysis Result Files” option from the main menu. Upon selecting the option from the main menu, the script displays a list of files available and the user can select the file to view. This is seen in the screenshot below. The source for this script can be seen in the Appendix and is titled *result\_files.sh*.



```
VM SNAPSHOT ANALYSIS

View Analysis Result Files

1) files_added.list
2) files_changed.list
3) files_deleted.list
4) files_edited.list
5) files_setid.list

0) EXIT

Please enter choice to view file [1-5,0]:
```

Figure 10: The View Analysis Result Files menu

#### 4.1.9 Compute MD5 & SHA1 hashes

The script allows users, as can be seen from the screenshot below, to compute the MD5 & SHA1 hashes of the files that have been selected for analysis or compute the MD5 & SHA1 hashes or hashes of any other file. This will allow the user to verify that the script is forensically sound and does not modify the evidence in any way. The source for this script can be seen in the Appendix and is titled *check\_md5.sh*.

```
VM SNAPSHOT ANALYSIS  
Compute MD5 & SHA1 Hashes  
1) Compute Hash for Files Already Selected  
2) Compute Hash for Another File  
0) EXIT  
Please enter choice [1-2,0]: █
```

Figure 11: Compute MD5 Hashes menu

## 5 Results

The Forensics Snapshot Analysis tool developed by the author is a powerful tool for forensics investigators to analyze snapshots of the same VM. This tool is also very useful in incident response to confirm a breach and to carry out further forensic analysis.

### 5.1 Image Acquisition

FTK and EnCase can natively handle VM hard disk (.vmdk) files. Both tools may require a write blocker device to image a physical drive. The author studied the functionality of both tools and concluded that when acquiring files instead of a physical drive, a write blocker need not be used. Since VMware virtual disks are implemented as files, they can be acquired without the use of a write blocker device using FTK and EnCase. To verify this hypothesis, the author used both the tools and checked for matching file hashes. Both tools produced matching MD5 and SHA1 hashes which verify vmdk files can be safely imaged using either tool with using a write block device (refer Figure 12 & 13). It is evident that the image is not affected while converting to raw/dd from vmdk, as images generated by FTK and EnCase have matching hashes.

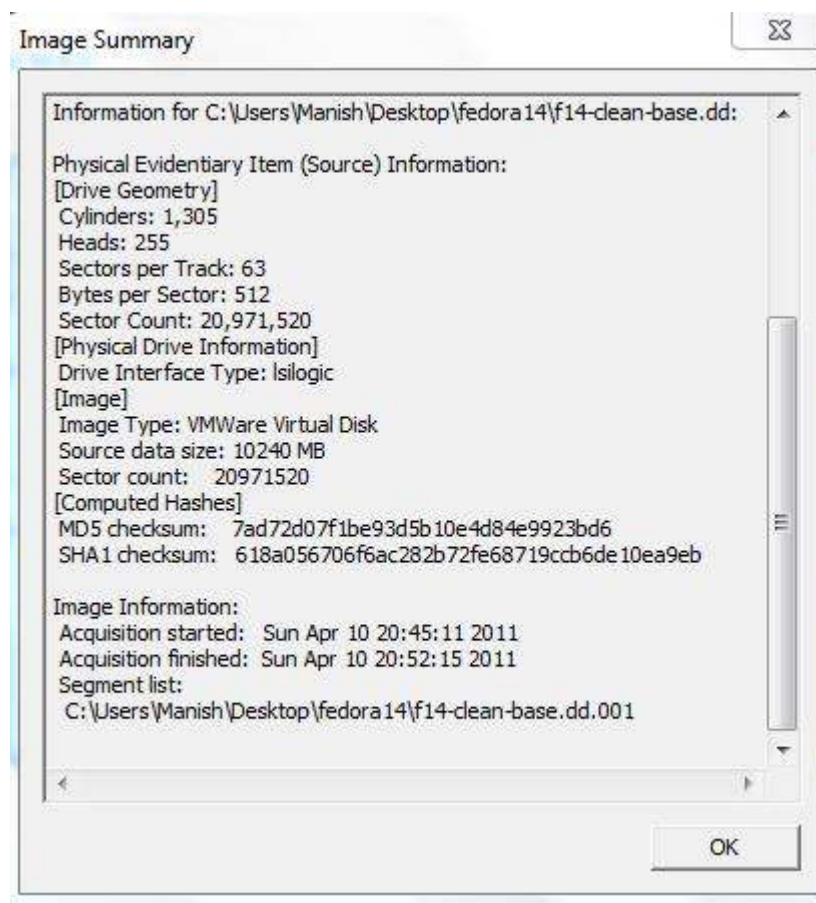


Figure 12: MD5 & SHA1 hashes generated after conversion to raw by FTK.

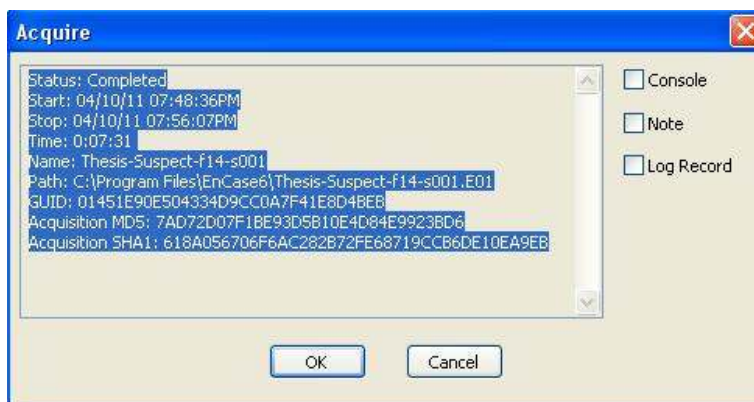


Figure 13: MD5 & SHA1 hashes generated after conversion to raw by EnCase.

## 5.2 Forensics Snapshot Analysis tool

The Forensics Snapshot Analysis tool successfully analyzes and compares snapshots of the same virtual machine taken at different points in time. Using the tool, a forensics examiner can generate a list of files that have been added, deleted, modified and changed by comparing the snapshots. The tool produces formatted reports of each analysis procedure it carries out. The forensics examiner can then decide on areas of further investigation based on the items of interest generated by the Forensics Snapshot Analysis tool.

The Forensics Snapshot Analysis tool is forensically sound and does not modify the raw files in any way. This can be proved by computing the hashes of the raw files after analysis is complete. The MD5 & SHA1 hashed computed for the raw files after the tool has analyzed the raw files matches the hashes computed before analysis was run (refer Figure 14). The hashes for all raw the files used can be found in the Appendix.

```

VM SNAPSHOT ANALYSIS
Compute MD5 & SHA1 Hashes
1) Compute Hash for Files Already Selected
2) Compute Hash for Another File
0) EXIT
Please enter choice [1-2,0]: 1
Computing MD5 & SHA1 for /home/manish/data/fl4-clean-flat-files/fl4-clean-flat-files-dd.001
Please be patient, this may take a while.
7ad72d07f1be93d5b10e4d84e9923bd6 /home/manish/data/fl4-clean-flat-files/fl4-clean-flat-files-dd.001
618a056706f6ac282b72fe68719ccb6de10ea9eb /home/manish/data/fl4-clean-flat-files/fl4-clean-flat-files-dd.001
Computing MD5 & SHA1 for /home/manish/data/fl4-changes-flat-files/fl4-changes-flat-files-dd.001
Please be patient, this may take a while.
a2977b599e0299d62d54f2c97cd4b824 /home/manish/data/fl4-changes-flat-files/fl4-changes-flat-files-dd.001
ab57296a20c0b2394518d334685817e208e621b9 /home/manish/data/fl4-changes-flat-files/fl4-changes-flat-files-dd.001
Please press <ENTER> to continue.

```

Figure 14: MD5 & SHA1 hashes obtained from the Forensics Analysis tool after analysis.

## 5.3 Forensics Snapshot Analysis Results

The setup of the Suspect Fedora VM and Analysis Fedora VM is described in section 3. Once the BASH script was completed, the author tested the script by having it analyze the snapshots obtained from the various stages of the Suspect Fedora VM. The script was successfully able to highlight files that were deleted, added, edited or changed in any way. The result files of the script can be seen in below.

### 5.3.1 Files Deleted

The list of files seen below, are the files that were deleted after the base/clean snapshot was taken.

Following are the files that have been deleted:

```
/etc/mtab~2122 (deleted)
/etc/systemd/system/graphical.target.wants/firstboot-graphical.service
(deleted)
/home/manish/Desktop/.added3.txt.swp (deleted)
/home/manish/Desktop/del_me.txt
/home/manish/.local/share/gnote/73b05277-b147-4cec-ad05-
31b957eba200.note
/home/manish/.local/share/gnote/73b05277-b147-4cec-ad05-
31b957eba200.note.tmp (deleted-realloc)
/home/manish/.local/share/gvfs-metadata/home-2a4bc7a9.log (deleted)
/home/manish/.local/share/gvfs-metadata/home-759096e1.log
/home/manish/.local/share/gvfs-metadata/home-759096e1.log.ZLIORV
(deleted-realloc)
/home/manish/.local/share/gvfs-metadata/home.8RVORV (deleted-realloc)
/home/manish/.mozilla/firefox/5ixj0qol.default/lock (deleted)
/home/manish/.mozilla/firefox/5ixj0qol.default/sessionstore.js
(deleted-realloc)
/$OrphanFiles/OrphanFile-174250 (deleted)
/$OrphanFiles/OrphanFile-181966 (deleted)
/root/.recently-used.xbel.VGYLRV (deleted-realloc)
/root/.xauthNrPy2u (deleted)
```

### 5.3.2 Files Added to Snapshot

The list of files seen below, are the files that were added after the base/clean snapshot was taken.

Following are the new files added:

```
/etc/mtab~2291 (deleted)
/etc/systemd/system/graphical.target.wants/firstboot-graphical.service
(deleted-realloc)
/home/manish/Desktop/added2.txt
/home/manish/Desktop/added.txt
/home/manish/Desktop/del_me2.txt (deleted)
/home/manish/.local/share/gnote/0e88dfa9-bea4-4644-a33d-
7ff8321abae4.note
```

```

/home/manish/.local/share/gnote/0e88dfa9-bea4-4644-a33d-
7ff8321abae4.note.tmp (deleted-realloc)
/home/manish/.local/share/gnote/Backup/73b05277-b147-4cec-ad05-
31b957eba200.note
/home/manish/.local/share/gvfs-metadata/home-3c629ae0.log
/home/manish/.local/share/gvfs-metadata/home-3c629ae0.log.QDWORV
(deleted-realloc)
/home/manish/.local/share/gvfs-metadata/home-759096e1.log (deleted)
/home/manish/.local/share/gvfs-metadata/home.W6IPRV (deleted-realloc)
/home/manish/.mozilla/firefox/5ixj0qol.default/lock (deleted-realloc)
/home/manish/.mozilla/firefox/5ixj0qol.default/sessionstore.js
(deleted)
/$OrphanFiles/OrphanFile-448723 (deleted)
/root/.viminfo.tmp (deleted-realloc)
/root/.xauthrbXq5g (deleted)
/root/.xauthrbXq5g-l (deleted)
/root/.xauthrbXq5g-n (deleted)

```

### 5.3.3 Files Edited

The list of files seen below, are the files that were edited after the base/clean snapshot was taken, i.e. files that had updated modification times (mtime). The file has been truncated in this section and the complete file can be found in the Appendix under *Result Files*.

Following are the files that have been edited:

```

Original Modification Time Sat 26 Feb 2011 06:53:50 PM EST
File name                  /etc
Current Modification Time  Sun 27 Feb 2011 12:15:16 AM EST

Original Modification Time Thu 24 Feb 2011 01:24:22 PM EST
File name                  /etc/hosts
Current Modification Time  Sun 27 Feb 2011 12:08:51 AM EST

Original Modification Time Sat 26 Feb 2011 06:53:50 PM EST
File name                  /etc/mtab
Current Modification Time  Sun 27 Feb 2011 12:15:16 AM EST

Original Modification Time Sat 26 Feb 2011 06:53:50 PM EST
File name                  /etc/mtab~ (deleted)
Current Modification Time  Sun 27 Feb 2011 12:15:16 AM EST

Original Modification Time Sat 26 Feb 2011 06:53:50 PM EST
File name                  /etc/mtab.tmp (deleted-realloc)
Current Modification Time  Sun 27 Feb 2011 12:15:16 AM EST

Original Modification Time Thu 24 Feb 2011 01:04:42 PM EST
File name                  /etc/rc.d
Current Modification Time  Sun 27 Feb 2011 12:08:04 AM EST

```

### 5.3.4 Files Changed

The list of files seen below, are the files that were changed after the base/clean snapshot was taken, i.e. files that had updated change times (ctime). The file has been truncated in this section and the complete file can be found in the Appendix under *Result Files*.

Following are the files that have been changed:

```
Original Change Time  Sat 26 Feb 2011 06:53:50 PM EST
File name            /etc
Current Change Time   Sun 27 Feb 2011 12:15:16 AM EST

Original Change Time  Thu 24 Feb 2011 01:24:22 PM EST
File name            /etc/hosts
Current Change Time   Sun 27 Feb 2011 12:08:51 AM EST

Original Change Time  Sat 26 Feb 2011 06:53:50 PM EST
File name            /etc/mtab
Current Change Time   Sun 27 Feb 2011 12:15:16 AM EST

Original Change Time  Sat 26 Feb 2011 06:53:50 PM EST
File name            /etc/mtab~ (deleted)
Current Change Time   Sun 27 Feb 2011 12:15:16 AM EST

Original Change Time  Sat 26 Feb 2011 06:53:50 PM EST
File name            /etc/mtab.tmp (deleted-realloc)
Current Change Time   Sun 27 Feb 2011 12:15:16 AM EST

Original Change Time  Thu 24 Feb 2011 01:04:42 PM EST
File name            /etc/rc.d
Current Change Time   Sun 27 Feb 2011 12:08:04 AM EST
```

### 5.3.5 Files SETUID/SETGID

The list of files seen below, are the files that were SETUID and/or SETGID after the base/clean snapshot was taken.

Following are the files with SETUID/SETGID changes.

```
/sbin/dosfslabel (deleted-realloc) | r/rrwsr-xr-x
/sbin/e2label | r/rrwsr-xr-x
/sbin/mkfs.cramfs (deleted-realloc) | r/rrwsr-xr-x
/sbin/modprobe | r/rrwsr-xr-x
/sbin/rmmod | r/rrwxr-sr-x
/sbin/service | r/rrwxr-sr-x
/sbin/tune2fs | r/rrwsr-xr-x
/usr/bin/nmap | r/rrwsr-xr-x
```

## ***5.4 Miscellaneous Observations***

### **5.4.1 Editing a File Changes its Inode**

When a file on the Linux system is edited using vi/vim, the inode of the file changes. The vi editor creates a temporary file while working on a file that is being modified by the user. Until a write instruction is sent, vi stores the information in this temporary file. Once the write instruction is sent, vi creates a new file with the same name and stores it in a different location. VI removes the original file and the temporary file after the write operations are complete.

If the hard link count to a file is more than one, the above behavior is not true. In this case, vi creates a temporary file to store data until a write instruction is sent and upon receiving a write instruction, vi writes to the original file and deletes the temporary file.

### **5.4.2 Inode Reallocation**

If a new file is created soon after deleting an existing file, the inode used by the deleted file is freed up and assigned to the new file. This result can be seen when the file del\_me.txt is deleted and added2.txt is created. The inode of file del\_me2.txt is assigned to added2.txt.

## ***5.5 Additional Uses for Tool***

The Forensics Snapshot Analysis tool can also be used for academic and training purposes. Students are taught that booting a suspect machine or VM that is shutdown or suspend can tamper evidence. Snapshots of a shutdown or suspended VM can be acquired before and after booting, and these snapshots can be analyzed using the tool developed, to practically demonstrate why booting is not a forensically sound procedure.



## **6 Limitations & Future Work**

This research is in no way comprehensive at this stage. Several avenues can be pursued as an extension to this research.

### ***6.1 Possible Methods of Obfuscation***

#### **6.1.1 MAC Times**

The Forensics Snapshot Analysis tool relies heavily on the MAC time of files to generate files of interest. A possible method for obfuscation could be the use of a tool that does not modify MAC times of files. TrueCrypt is one such tool. When a TrueCrypt volume is modified, only the change time is updated, whereas the modification time (mtime) and access time (atime) remains the same. The author used TrueCrypt with his script to study this behavior of TrueCrypt and added features to the script which now checks for change time (ctime) difference between the files found in both snapshots.

#### **6.1.2 Encryption**

The Forensics Snapshot Analysis tool cannot view the contents of files that have been encrypted. The tool will still list files which have differing modification & change times but the examiner will not be able to view the contents of the file, if it is encrypted. A method to analyze encrypted files and volumes can be developed and incorporated in the tool.

### ***6.2 Future Work***

Currently the tool only supports ext2 & ext3 file systems. Changes can be made to the developed tool to handle other file systems such as ext4, FAT and NTFS.

Autopsy is a web based tool which uses Sleuthkit as a backend. Autopsy uses Perl modules; since this tool utilizes binaries from Sleuthkit, this tool can be converted to Perl and can be used as module that can be integrated with Autopsy.

## 7 References

- [1] Access Data (2010). Forensics ToolKit. Retrieved from <http://www.accessdata.com/forensictoolkit.html>
- [2] Bares, R. (2009). Hiding in a virtual world using unconventionally installed operating systems. IEEE International Conference on Intelligence and Security Informatics. Dallas, TX.
- [3] Barrett, D., Kipper, G., (2010). Investigating Dead Virtual Environments, Virtualization and Forensics, Syngress, Boston, 2010, Pages 83-107
- [4] Beek, C. (2010). Virtual Forensics. Retrieved from: [http://securitybananas.com/wp-content/uploads/2010/04/Virtual-Forensics\\_BlackHatEurope2010\\_CB.pdf](http://securitybananas.com/wp-content/uploads/2010/04/Virtual-Forensics_BlackHatEurope2010_CB.pdf)
- [5] Brown, C. L. T. (2005). Computer Evidence: Collection & Preservation. Hingham, MA: Charles River Media.
- [6] Ebaca (2010). Penguin Sleuth Kit Virtual Computer Forensics and Security Platform. Retrieved from <http://www.vmware.com/appliances/directory/249>.
- [7] E-fense (2010). Cyber Security & Computer Forensics Software Page. Retrieved from <http://www.e-fense.com/helix/>
- [8] Fiterman, E. M., & Durick, J. D. (2010). Ghost in the machine: Forensic evidence collection in the virtual environment. Digital Forensics Magazine, 2, 73–77.
- [9] Guidance Software. (2010). EnCase Forensic. Retrieved from: <http://www.guidancesoftware.com/forensic.htm>
- [10] Kruse, W. G., & Heiser, J. G. (2002). Computer Forensics: Incident Response Essentials (1st ed.): Addison Wesley Professional.
- [11] Metasploit (2010), Metasploit Anti-Forensics Project. Retrieved from <http://www.metasploit.com/research/projects/antiforensics/>
- [13] Messmer, E. (2009, October 20). Gartner predicts nearly half of server workloads will be virtualized. Network World. Retrieved from <http://www.networkworld.com/news/2009/102009-gartner-server-virtualization.html>
- [14] Mrdovic, S., Huseinovic, A., Zajko, E. (2009). Combining static and live digital forensic analysis in virtual environment. Information, Communication and Automation Technologies, 2009. ICAT 2009. XXII International Symposium on, vol., no., pp.1-6, 29-31 Oct.2009.
- [15] Nelson, B., Phillips, A., Enfinger, F., & Steuart, C. (2006). Guide to Computer Forensics and Investigations, Second Edition. Boston, MA: Thomson Course Technology.
- [16] nmap.org (2011). Nmap Retrieved from <http://www.nmap.org/>
- [17] Rude, T. (2000). DD and Computer Forensics. Retrieved <http://www.crazytrain.com/dd.html>

[18] sleuthkit.org (2010). Sleuthkit. Retrieved from <http://www.sleuthkit.org/sleuthkit/>

[19] tucrypt.org (2011). TureCrypt. Retrieved from <http://www.truecrypt.org/>

[20] VMware (2010). What Files Make Up a Virtual Machine?  
[http://www.vmware.com/support/ws55/doc/ws\\_learning\\_files\\_in\\_a\\_vm.html](http://www.vmware.com/support/ws55/doc/ws_learning_files_in_a_vm.html)

## Appendix

### ***MD5 & SHA1 Hashes***

Below are the MD5 & SHA1 hashes for the base/clean and the unclean/compromised system. These hashes are obtained from FTK, EnCase and the Forensics Snapshot Analysis tool.

#### **Base/Clean Snapshot**

MD5 checksum: 7ad72d07f1be93d5b10e4d84e9923bd6

SHA1 checksum: 618a056706f6ac282b72fe68719ccb6de10ea9eb

#### **Unclean/Compromised Snapshot**

MD5 checksum: a2977b599e0299d62d54f2c97cd4b024

SHA1 checksum: ab57296a20c0b2394518d334685817e208e621b9

## ***Result Files***

### **Files Edited**

Following are the files that have been edited:

```
Original Modification Time Sat 26 Feb 2011 06:53:50 PM EST
File name                  /etc
Current Modification Time  Sun 27 Feb 2011 12:15:16 AM EST

Original Modification Time Thu 24 Feb 2011 01:24:22 PM EST
File name                  /etc/hosts
Current Modification Time  Sun 27 Feb 2011 12:08:51 AM EST

Original Modification Time Sat 26 Feb 2011 06:53:50 PM EST
File name                  /etc/mtab
Current Modification Time  Sun 27 Feb 2011 12:15:16 AM EST

Original Modification Time Sat 26 Feb 2011 06:53:50 PM EST
File name                  /etc/mtab~ (deleted)
Current Modification Time  Sun 27 Feb 2011 12:15:16 AM EST

Original Modification Time Sat 26 Feb 2011 06:53:50 PM EST
File name                  /etc/mtab.tmp (deleted-realloc)
Current Modification Time  Sun 27 Feb 2011 12:15:16 AM EST

Original Modification Time Thu 24 Feb 2011 01:04:42 PM EST
File name                  /etc/rc.d
Current Modification Time  Sun 27 Feb 2011 12:08:04 AM EST

Original Modification Time Thu 24 Feb 2011 01:04:42 PM EST
File name                  /etc/rc.d/rc.local
Current Modification Time  Sun 27 Feb 2011 12:08:04 AM EST

Original Modification Time Sat 26 Feb 2011 06:53:50 PM EST
File name                  /etc/rc.d/.rc.local.swp (deleted-realloc)
Current Modification Time  Sun 27 Feb 2011 12:08:51 AM EST

Original Modification Time Sat 26 Feb 2011 06:53:50 PM EST
File name                  /etc/rc.d/.rc.local.swx (deleted)
Current Modification Time  Sun 27 Feb 2011 12:15:16 AM EST

Original Modification Time Sat 26 Feb 2011 06:47:30 PM EST
File name                  /etc/resolv.conf
Current Modification Time  Sun 27 Feb 2011 12:08:51 AM EST

Original Modification Time Thu 24 Feb 2011 01:04:34 PM EST
File name                  /etc/selinux
Current Modification Time  Sun 27 Feb 2011 12:07:27 AM EST
```

Original Modification Time Thu 24 Feb 2011 01:04:34 PM EST  
File name /etc/selinux/config  
Current Modification Time Sun 27 Feb 2011 12:07:27 AM EST

Original Modification Time Sat 26 Feb 2011 06:53:50 PM EST  
File name /etc/selinux/config~ (deleted)  
Current Modification Time Sun 27 Feb 2011 12:15:16 AM EST

Original Modification Time Sat 26 Feb 2011 06:53:50 PM EST  
File name /etc/selinux/.config.swp (deleted-realloc)  
Current Modification Time Sun 27 Feb 2011 12:08:51 AM EST

Original Modification Time Sat 26 Feb 2011 06:47:30 PM EST  
File name /etc/systemd/system/multi-  
user.target.wants/firstboot-text.service (deleted-realloc)  
Current Modification Time Sun 27 Feb 2011 12:08:51 AM EST

Original Modification Time Sat 26 Feb 2011 06:50:51 PM EST  
File name /home/manish/Desktop  
Current Modification Time Sun 27 Feb 2011 12:10:43 AM EST

Original Modification Time Sat 26 Feb 2011 06:47:36 PM EST  
File name /home/manish/.local/share/gnote  
Current Modification Time Sun 27 Feb 2011 12:07:04 AM EST

Original Modification Time Sat 26 Feb 2011 06:47:32 PM EST  
File name /home/manish/.local/share/gnote/Backup  
Current Modification Time Sun 27 Feb 2011 12:07:00 AM EST

Original Modification Time Sat 26 Feb 2011 06:51:53 PM EST  
File name /home/manish/.local/share/gvfs-metadata  
Current Modification Time Sun 27 Feb 2011 12:10:39 AM EST

Original Modification Time Sat 26 Feb 2011 06:51:53 PM EST  
File name /home/manish/.local/share/gvfs-metadata/home  
Current Modification Time Sun 27 Feb 2011 12:10:39 AM EST

Original Modification Time Sat 26 Feb 2011 06:52:28 PM EST  
File name /lib/ld-2.12.90.so (deleted-realloc)  
Current Modification Time Sun 27 Feb 2011 12:14:32 AM EST

Original Modification Time Sat 26 Feb 2011 06:52:28 PM EST  
File name /lib/libcrypt-2.12.90.so (deleted)  
Current Modification Time Sun 27 Feb 2011 12:14:32 AM EST

Original Modification Time Sat 26 Feb 2011 06:53:49 PM EST  
File name /media  
Current Modification Time Sun 27 Feb 2011 12:15:15 AM EST

Original Modification Time Sat 26 Feb 2011 06:53:49 PM EST  
File name /media/truecrypt1 (deleted)  
Current Modification Time Sun 27 Feb 2011 12:15:15 AM EST

Original Modification Time Sat 26 Feb 2011 06:52:28 PM EST  
File name /\$OrphanFiles/OrphanFile-18202 (deleted)  
Current Modification Time Sun 27 Feb 2011 12:15:08 AM EST

Original Modification Time Sat 26 Feb 2011 06:54:07 PM EST  
File name /root  
Current Modification Time Sun 27 Feb 2011 12:15:18 AM EST

Original Modification Time Sat 26 Feb 2011 06:54:07 PM EST  
File name /root/.bash\_history  
Current Modification Time Sun 27 Feb 2011 12:15:11 AM EST

Original Modification Time Sat 26 Feb 2011 06:53:52 PM EST  
File name /root/.TrueCrypt/Configuration.xml  
Current Modification Time Sun 27 Feb 2011 12:15:18 AM EST

Original Modification Time Sat 26 Feb 2011 06:53:52 PM EST  
File name /root/.TrueCrypt-lock-root (deleted)  
Current Modification Time Sun 27 Feb 2011 12:15:18 AM EST

Original Modification Time Sat 26 Feb 2011 06:47:44 PM EST  
File name /root/.viminfo  
Current Modification Time Sun 27 Feb 2011 12:15:08 AM EST

Original Modification Time Sat 26 Feb 2011 06:53:50 PM EST  
File name /tmp  
Current Modification Time Sun 27 Feb 2011 12:15:16 AM EST

Original Modification Time Sat 26 Feb 2011 06:48:52 PM EST  
File name /tmp/pulse-PKdhtXMmr18n/native (deleted-realloc)  
Current Modification Time Sun 27 Feb 2011 12:13:39 AM EST

Original Modification Time Sat 26 Feb 2011 06:53:50 PM EST  
File name /tmp/.truecrypt\_aux\_mnt1 (deleted)  
Current Modification Time Sun 27 Feb 2011 12:15:16 AM EST

Original Modification Time Sat 26 Feb 2011 06:52:35 PM EST  
File name /tmp/vteA88PRV (deleted-realloc)  
Current Modification Time Sat 26 Feb 2011 06:55:45 PM EST

Original Modification Time Sat 26 Feb 2011 06:47:44 PM EST  
File name /usr/libexec/utempter/utempter;4d59fa75 (deleted-realloc)  
Current Modification Time Sun 27 Feb 2011 12:15:08 AM EST

Original Modification Time Mon 04 Oct 2010 07:31:43 AM EDT  
File name /usr/lib/gimp/2.0/plugin-ins/gee (deleted-realloc)  
Current Modification Time Mon 04 Oct 2010 07:31:51 AM EDT

Original Modification Time Tue 22 Jun 2010 11:15:21 AM EDT  
File name /usr/share/man/man1/..1.gz

Current Modification Time Tue 05 Oct 2010 06:41:58 PM EDT

Original Modification Time Thu 24 Feb 2011 01:24:21 PM EST

File name /var/lib/dhclient/dhclient-5fb06bd0-0bb0-7ffb-45f1-d6edd65f3e03-eth0.lease

Current Modification Time Sun 27 Feb 2011 12:08:50 AM EST

Original Modification Time Sat 26 Feb 2011 06:53:16 PM EST

File name /var/lib/PackageKit

Current Modification Time Sun 27 Feb 2011 12:13:19 AM EST

Original Modification Time Sat 26 Feb 2011 06:53:16 PM EST

File name /var/lib/PackageKit/transactions.db

Current Modification Time Sun 27 Feb 2011 12:13:19 AM EST

Original Modification Time Sat 26 Feb 2011 06:53:16 PM EST

File name /var/lib/PackageKit/transactions.db-journal  
(deleted)

Current Modification Time Sun 27 Feb 2011 12:13:19 AM EST



## Files Changed

Following are the files that have been changed:

```
Original Change Time  Sat 26 Feb 2011 06:53:50 PM EST
File name             /etc
Current Change Time   Sun 27 Feb 2011 12:15:16 AM EST

Original Change Time  Thu 24 Feb 2011 01:24:22 PM EST
File name             /etc/hosts
Current Change Time   Sun 27 Feb 2011 12:08:51 AM EST

Original Change Time  Sat 26 Feb 2011 06:53:50 PM EST
File name             /etc/mtab
Current Change Time   Sun 27 Feb 2011 12:15:16 AM EST

Original Change Time  Sat 26 Feb 2011 06:53:50 PM EST
File name             /etc/mtab~ (deleted)
Current Change Time   Sun 27 Feb 2011 12:15:16 AM EST

Original Change Time  Sat 26 Feb 2011 06:53:50 PM EST
File name             /etc/mtab.tmp (deleted-realloc)
Current Change Time   Sun 27 Feb 2011 12:15:16 AM EST

Original Change Time  Thu 24 Feb 2011 01:04:42 PM EST
File name             /etc/rc.d
Current Change Time   Sun 27 Feb 2011 12:08:04 AM EST

Original Change Time  Thu 24 Feb 2011 01:04:42 PM EST
File name             /etc/rc.d/rc.local
Current Change Time   Sun 27 Feb 2011 12:08:04 AM EST

Original Change Time  Sat 26 Feb 2011 06:53:50 PM EST
File name             /etc/rc.d/.rc.local.swp (deleted-realloc)
Current Change Time   Sun 27 Feb 2011 12:08:51 AM EST

Original Change Time  Sat 26 Feb 2011 06:53:50 PM EST
File name             /etc/rc.d/.rc.local.swx (deleted)
Current Change Time   Sun 27 Feb 2011 12:15:16 AM EST

Original Change Time  Sat 26 Feb 2011 06:47:30 PM EST
File name             /etc/resolv.conf
Current Change Time   Sun 27 Feb 2011 12:08:51 AM EST

Original Change Time  Thu 24 Feb 2011 01:04:34 PM EST
File name             /etc/selinux
Current Change Time   Sun 27 Feb 2011 12:07:27 AM EST

Original Change Time  Thu 24 Feb 2011 01:04:34 PM EST
File name             /etc/selinux/config
```

Current Change Time Sun 27 Feb 2011 12:07:27 AM EST

Original Change Time Sat 26 Feb 2011 06:53:50 PM EST  
File name /etc/selinux/config~ (deleted)  
Current Change Time Sun 27 Feb 2011 12:15:16 AM EST

Original Change Time Sat 26 Feb 2011 06:53:50 PM EST  
File name /etc/selinux/.config.swp (deleted-realloc)  
Current Change Time Sun 27 Feb 2011 12:08:51 AM EST

Original Change Time Sat 26 Feb 2011 06:47:30 PM EST  
File name /etc/systemd/system/multi-user.target.wants/firstboot-text.service (deleted-realloc)  
Current Change Time Sun 27 Feb 2011 12:08:51 AM EST

Original Change Time Sat 26 Feb 2011 06:50:51 PM EST  
File name /home/manish/Desktop  
Current Change Time Sun 27 Feb 2011 12:10:43 AM EST

Original Change Time Sat 26 Feb 2011 06:53:50 PM EST  
File name /home/manish/Desktop/true\_crypt.txt  
Current Change Time Sun 27 Feb 2011 12:15:16 AM EST

Original Change Time Sat 26 Feb 2011 06:47:36 PM EST  
File name /home/manish/.local/share/gnote  
Current Change Time Sun 27 Feb 2011 12:07:04 AM EST

Original Change Time Sat 26 Feb 2011 06:47:32 PM EST  
File name /home/manish/.local/share/gnote/Backup  
Current Change Time Sun 27 Feb 2011 12:07:00 AM EST

Original Change Time Sat 26 Feb 2011 06:51:53 PM EST  
File name /home/manish/.local/share/gvfs-metadata  
Current Change Time Sun 27 Feb 2011 12:10:39 AM EST

Original Change Time Sat 26 Feb 2011 06:51:53 PM EST  
File name /home/manish/.local/share/gvfs-metadata/home  
Current Change Time Sun 27 Feb 2011 12:10:39 AM EST

Original Change Time Sat 26 Feb 2011 06:52:28 PM EST  
File name /lib/ld-2.12.90.so (deleted-realloc)  
Current Change Time Sun 27 Feb 2011 12:14:32 AM EST

Original Change Time Sat 26 Feb 2011 06:52:28 PM EST  
File name /lib/libcrypt-2.12.90.so (deleted)  
Current Change Time Sun 27 Feb 2011 12:14:32 AM EST

Original Change Time Sat 26 Feb 2011 06:53:49 PM EST  
File name /media  
Current Change Time Sun 27 Feb 2011 12:15:15 AM EST

Original Change Time Sat 26 Feb 2011 06:53:49 PM EST

```
File name          /media/truecrypt1 (deleted)
Current Change Time    Sun 27 Feb 2011 12:15:15 AM EST

Original Change Time  Sat 26 Feb 2011 06:52:28 PM EST
File name            /$OrphanFiles/OrphanFile-18202 (deleted)
Current Change Time   Sun 27 Feb 2011 12:15:08 AM EST

Original Change Time  Sat 26 Feb 2011 06:54:07 PM EST
File name            /root
Current Change Time   Sun 27 Feb 2011 12:15:18 AM EST

Original Change Time  Sat 26 Feb 2011 06:54:07 PM EST
File name            /root/.bash_history
Current Change Time   Sun 27 Feb 2011 12:15:11 AM EST

Original Change Time  Sat 26 Feb 2011 06:50:15 PM EST
File name            /root/.config/ibus/bus
Current Change Time   Sun 27 Feb 2011 12:13:46 AM EST

Original Change Time  Sat 26 Feb 2011 06:53:52 PM EST
File name            /root/.TrueCrypt/Configuration.xml
Current Change Time   Sun 27 Feb 2011 12:15:18 AM EST

Original Change Time  Sat 26 Feb 2011 06:53:52 PM EST
File name            /root/.TrueCrypt-lock-root (deleted)
Current Change Time   Sun 27 Feb 2011 12:15:18 AM EST

Original Change Time  Sat 26 Feb 2011 06:47:44 PM EST
File name            /root/.viminfo
Current Change Time   Sun 27 Feb 2011 12:15:08 AM EST

Original Change Time  Thu 24 Feb 2011 01:09:49 PM EST
File name            /sbin/dosfslabel (deleted-realloc)
Current Change Time   Sun 27 Feb 2011 12:08:59 AM EST

Original Change Time  Thu 24 Feb 2011 01:08:49 PM EST
File name            /sbin/e2label
Current Change Time   Sun 27 Feb 2011 12:08:49 AM EST

Original Change Time  Thu 24 Feb 2011 01:08:49 PM EST
File name            /sbin/mkfs.cramfs (deleted-realloc)
Current Change Time   Sun 27 Feb 2011 12:08:49 AM EST

Original Change Time  Thu 24 Feb 2011 01:09:49 PM EST
File name            /sbin/modprobe
Current Change Time   Sun 27 Feb 2011 12:08:59 AM EST

Original Change Time  Thu 24 Feb 2011 01:09:42 PM EST
File name            /sbin/rmmod
Current Change Time   Sun 27 Feb 2011 12:09:06 AM EST

Original Change Time  Thu 24 Feb 2011 01:06:17 PM EST
```

```
File name          /sbin/service
Current Change Time    Sun 27 Feb 2011 12:08:40 AM EST

Original Change Time   Thu 24 Feb 2011 01:08:49 PM EST
File name             /sbin/tune2fs
Current Change Time    Sun 27 Feb 2011 12:08:49 AM EST

Original Change Time   Sat 26 Feb 2011 06:53:50 PM EST
File name             /tmp
Current Change Time    Sun 27 Feb 2011 12:15:16 AM EST

Original Change Time   Sat 26 Feb 2011 06:48:52 PM EST
File name             /tmp/pulse-PKdhtXMmr18n/native (deleted-realloc)
Current Change Time    Sun 27 Feb 2011 12:13:39 AM EST

Original Change Time   Sat 26 Feb 2011 06:53:50 PM EST
File name             /tmp/.truecrypt_aux_mnt1 (deleted)
Current Change Time    Sun 27 Feb 2011 12:15:16 AM EST

Original Change Time   Sat 26 Feb 2011 06:52:35 PM EST
File name             /tmp/vteA88PRV (deleted-realloc)
Current Change Time    Sat 26 Feb 2011 06:55:45 PM EST

Original Change Time   Thu 24 Feb 2011 01:05:30 PM EST
File name             /usr/bin/nmap
Current Change Time    Sun 27 Feb 2011 12:08:21 AM EST

Original Change Time   Sat 26 Feb 2011 06:47:44 PM EST
File name             /usr/libexec/utempter/utempter;4d59fa75 (deleted-realloc)
Current Change Time    Sun 27 Feb 2011 12:15:08 AM EST

Original Change Time   Thu 24 Feb 2011 12:28:35 PM EST
File name             /usr/lib/gimp/2.0/plugin-ins/gee (deleted-realloc)
Current Change Time    Thu 24 Feb 2011 12:29:28 PM EST

Original Change Time   Mon 14 Feb 2011 11:01:49 PM EST
File name             /usr/share/man/man1/..1.gz
Current Change Time    Mon 14 Feb 2011 11:03:35 PM EST

Original Change Time   Thu 24 Feb 2011 01:24:21 PM EST
File name             /var/lib/dhclient/dhclient-5fb06bd0-0bb0-7ffb-45f1-d6edd65f3e03-eth0.lease
Current Change Time    Sun 27 Feb 2011 12:08:50 AM EST

Original Change Time   Sat 26 Feb 2011 06:53:16 PM EST
File name             /var/lib/PackageKit
Current Change Time    Sun 27 Feb 2011 12:13:19 AM EST

Original Change Time   Sat 26 Feb 2011 06:53:16 PM EST
File name             /var/lib/PackageKit/transactions.db
Current Change Time    Sun 27 Feb 2011 12:13:19 AM EST
```

Original Change Time Sat 26 Feb 2011 06:53:16 PM EST  
File name /var/lib/PackageKit/transactions.db-journal (deleted)  
Current Change Time Sun 27 Feb 2011 12:13:19 AM EST

Original Change Time Sat 26 Feb 2011 06:54:07 PM EST  
File name /var/log/audit/audit.log  
Current Change Time Sun 27 Feb 2011 12:15:11 AM EST

Original Change Time Thu 24 Feb 2011 01:24:24 PM EST  
File name /var/log/maillog  
Current Change Time Sun 27 Feb 2011 12:14:17 AM EST

Original Change Time Sat 26 Feb 2011 06:55:00 PM EST  
File name /var/log/messages  
Current Change Time Sun 27 Feb 2011 12:13:17 AM EST

Original Change Time Sat 26 Feb 2011 06:54:07 PM EST  
File name /var/log/secure  
Current Change Time Sun 27 Feb 2011 12:15:11 AM EST

Original Change Time Thu 24 Feb 2011 01:24:21 PM EST  
File name /var/run/dhclient-eth0.pid  
Current Change Time Sun 27 Feb 2011 12:08:50 AM EST

Original Change Time Sat 26 Feb 2011 06:52:28 PM EST  
File name /var/run/gdm/auth-for-manish-ZPwM9i  
Current Change Time Sun 27 Feb 2011 12:14:32 AM EST

Original Change Time Sat 26 Feb 2011 06:52:28 PM EST  
File name /var/run/gdm/auth-for-manish-ZPwM9i/database-c  
(deleted)  
Current Change Time Sun 27 Feb 2011 12:14:32 AM EST

Original Change Time Sat 26 Feb 2011 06:52:28 PM EST  
File name /var/run/gdm/auth-for-manish-ZPwM9i/database-l  
(deleted)  
Current Change Time Sun 27 Feb 2011 12:14:32 AM EST

Original Change Time Sat 26 Feb 2011 06:53:21 PM EST  
File name /var/run/nm-dhclient-eth1.conf (deleted)  
Current Change Time Sun 27 Feb 2011 12:13:19 AM EST

Original Change Time Sat 26 Feb 2011 06:53:21 PM EST  
File name /var/run/yum.pid (deleted)  
Current Change Time Sun 27 Feb 2011 12:13:19 AM EST

## ***Script Source Code***

### **menu.sh**

```
#!/bin/bash

function CTRAP {
    trap '
        let lne++
        let lne++
        tput cup $lne $cl
        echo -n "CTRL_C Not Allowed."
        let lne++
        tput cup $lne $cl
        echo -n "Please press <ENTER> to continue."
    ' 2
}

MENUPATH=/home/manish/snapshot_diff
OUTPUTDIR=${MENUPATH}/output_files
TEMP_CLEAN=/tmp/clean
TEMP_CHANGED=/tmp/changed
DATAFILE1=${MENUPATH}/raw_files_path
DATAFILE_CLEAN=${MENUPATH}/files_list_clean
DATAFILE_CHANGED=${MENUPATH}/files_list_changed
OUTPUT_ADDED=${OUTPUTDIR}/files_added.list
OUTPUT_DELETED=${OUTPUTDIR}/files_deleted.list
OUTPUT_EDITED=${OUTPUTDIR}/files_edited.list
OUTPUT_CHANGED=${OUTPUTDIR}/files_changed.list
OUTPUT_SETID=${OUTPUTDIR}/files_setid.list

choice=1
until [ $choice == 0 ]
do
    lne=5
    cl=5
    tput clear
    tput smso
    tput cup $lne $cl
    echo -n "VM SNAPSHOT ANALYSIS"
    tput rmso
    lne=`expr $lne + 2`
    tput cup $lne $cl
    echo -n "1) Select Snapshots to Compare"
    let lne++
    tput cup $lne $cl
    echo -n "2) View Selected Snapshots"
    let lne++
    tput cup $lne $cl
    echo -n "3) View Files Deleted"
    let lne++

```

```

tput cup $lne $cl
echo -n "4) View New Files Added"
let lne++
tput cup $lne $cl
echo -n "5) View Files Edited [Modification Time]"
let lne++
tput cup $lne $cl
echo -n "6) View Files Changed [Change Time]"
let lne++
tput cup $lne $cl
echo -n "7) View SETUID/SETGUID Changes"
let lne++
tput cup $lne $cl
echo -n "8) View Analysis Result Files"
let lne++
tput cup $lne $cl
echo -n "9) Compute MD5 & SHA1 Hashes"
lne=`expr $lne + 2`
tput cup $lne $cl
echo -n "0) EXIT"
lne=`expr $lne + 2`
tput cup $lne $cl
echo -n "Please enter choice [1-9,0]: "
CTRAP; read choice
case $choice in
    1) $MENUPATH/select_files.sh
        ;;
    2) $MENUPATH/view_files.sh
        ;;
    3) $MENUPATH/deleted_files.sh
        ;;
    4) $MENUPATH/added_files.sh
        ;;
    5) $MENUPATH/edited_files.sh
        ;;
    6) $MENUPATH/change_files.sh
        ;;
    7) $MENUPATH/setid_changes.sh
        ;;
    8) $MENUPATH/result_files.sh
        ;;
    9) $MENUPATH/check_md5.sh
        ;;
    0) echo ; echo ; exit
        ;;
    *)
        lne=`expr $lne + 2`
        tput cup $lne $cl
        echo -n "Choice Invalid."
        let lne++
        tput cup $lne $cl
        echo -n "Please press <ENTER> to continue."

```

```
done      esac      ;;      read
```



**select\_files.sh**

```
#!/bin/bash

function CTRAP {
    trap '
        let lne++
        let lne++
        tput cup $lne $cl
        echo -n "CTRL_C Not Allowed."
        let lne++
        tput cup $lne $cl
        echo -n "Please press <ENTER> to continue."
        ' 2
}

MENUPATH=/home/manish/snapshot_diff
OUTPUTDIR=${MENUPATH}/output_files
TEMP_CLEAN=/tmp/clean
TEMP_CHANGED=/tmp/changed
DATAFILE1=${MENUPATH}/raw_files_path
DATAFILE_CLEAN=${MENUPATH}/files_list_clean
DATAFILE_CHANGED=${MENUPATH}/files_list_changed
OUTPUT_ADDED=${OUTPUTDIR}/files_added.list
OUTPUT_DELETED=${OUTPUTDIR}/files_deleted.list
OUTPUT_EDITED=${OUTPUTDIR}/files_edited.list
OUTPUT_SETID=${OUTPUTDIR}/files_setid.list

lne=5
cl=5
CTRAP
tput clear
tput smso
tput cup $lne $cl
echo -n "VM SNAPSHOT ANALYSIS"
lne=`expr $lne + 2`
tput cup $lne $cl
echo -n "Select Snapshots to Compare"
tput rmso
lne=`expr $lne + 2`
tput cup $lne $cl
echo -n "Note 1: Please input path of RAW files only."
lne=`expr $lne + 2`
tput cup $lne $cl
echo -n "Note 2: Please input complete file path."
lne=`expr $lne + 2`
tput cup $lne $cl
echo -n "Note 3: Please make sure the snapshots belong to the
same VM."
```

```

function PATH_1 {
    line=`expr $line + 2`
    tput cup $line $cl
    echo -n "Please input path of clean snapshot which "
    let line++
    tput cup $line $cl
    echo -n "will be used as a baseline for comparison."
    let line++
    tput cup $line $cl
    echo -n "PATH 1: "
    read path1
    if [ -z $path1 ]; then
        let line++
        tput cup $line $cl
        echo -n "Path cannot be left empty."
        let line++
        tput cup $line $cl
        echo -n "Do you wish to continue (y/n): "
        let line++
        tput cup $line $cl
        echo -n "Enter N/n to exit."
        line=`expr $line - 1`
        tput cup $line 36
        read ans
        if [ $ans == n -o $ans == N ] &> /dev/null ; then
            exit
        else
            PATH_1
        fi
    else
        if [ -f $path1 ]; then
            echo $path1 > $DATAFILE1
        else
            let line++
            tput cup $line $cl
            echo -n "File does not exist or path incorrect."
            let line++
            tput cup $line $cl
            echo -n "Do you wish to continue (y/n): "
            let line++
            tput cup $line $cl
            echo -n "Enter N/n to exit."
            line=`expr $line - 1`
            tput cup $line 36
            read ans
            if [ $ans == n -o $ans == N ] &> /dev/null ; then
                exit
            else
                PATH_1
            fi
        fi
    fi
fi

```

```

}
PATH_1

function PATH_2 {
    line=`expr $line + 2`
    tput cup $line $cl
    echo -n "Please input path of suspected snapshot which"
    let line++
    tput cup $line $cl
    echo -n "will be against the baseline for comparison."
    let line++
    tput cup $line $cl
    echo -n "PATH 2: "
    read path2
    if [ -z $path2 ]; then
        let line++
        tput cup $line $cl
        echo -n "Path cannot be left empty."
        let line++
        tput cup $line $cl
        echo -n "Do you wish to continue (y/n): "
        let line++
        tput cup $line $cl
        echo -n "Enter N/n to exit."
        line=`expr $line - 1`
        tput cup $line 36
        read ans
        if [ $ans == n -o $ans == N ] &> /dev/null ; then
            exit
        else
            PATH_2
        fi
    else
        if [ -f $path2 ]; then
            echo $path2 >> $DATAFILE1
        else
            let line++
            tput cup $line $cl
            echo -n "File does not exist or path incorrect."
            let line++
            tput cup $line $cl
            echo -n "Do you wish to continue (y/n): "
            let line++
            tput cup $line $cl
            echo -n "Enter N/n to exit."
            line=`expr $line - 1`
            tput cup $line 36
            read ans
            if [ $ans == n -o $ans == N ] &> /dev/null ; then
                exit
            else
                PATH_2
            fi
        fi
    fi
}

```

```

        fi
    fi
}
PATH_2

function SKEW {
    lne=`expr $lne + 2`
    tput cup $lne $cl
    echo -n "Please input the time skew of the system in
seconds, else enter 0:"
    read skew
    if [ -z $skew ]; then
        let lne++
        tput cup $lne $cl
        echo -n "Skew cannot be left empty."
        let lne++
        tput cup $lne $cl
        echo -n "Do you wish to continue (y/n): "
        let lne++
        tput cup $lne $cl
        echo -n "Enter N/n to exit."
        line=`expr $lne - 1`
        tput cup $line 36
        read ans
        if [ $ans == n -o $ans == N ] &> /dev/null ; then
            exit
        else
            SKEW
        fi
    else
        skew1=`echo $skew | sed 's/-//`
        if [[ $skew1 == *([0-9]) ]]; then
            echo $skew >> $DATAFILE1
            lne=`expr $lne + 2`
            tput cup $lne $cl
            echo -n "Both files seem to be valid, paths
recorded."

            let lne++
            tput cup $lne $cl
            echo -n "Please press <ENTER> to continue."
            read
        else
            let lne++
            tput cup $lne $cl
            echo -n "Input is not a number."
            let lne++
            tput cup $lne $cl
            echo -n "Do you wish to continue (y/n): "
            let lne++
            tput cup $lne $cl
            echo -n "Enter N/n to exit."
        fi
    fi
}

```

```
        line=`expr $line - 1`  
        tput cup $line 36  
        read ans  
        if [ $ans == n -o $ans == N ] &> /dev/null ; then  
            exit  
        else  
            SKEW  
        fi  
    fi  
fi  
}  
SKEW
```

**view\_files.sh**

```
#!/bin/bash

function CTRAP {
    trap '
        let lne++
        let lne++
        tput cup $lne $cl
        echo -n "CTRL_C Not Allowed."
        let lne++
        tput cup $lne $cl
        echo -n "Please press <ENTER> to continue."
    ' 2
}

function NO_FILE {
    lne=`expr $lne + 2`
    tput cup $lne $cl
    echo -n "No snapshot files selected."
}

MENUPATH=/home/manish/snapshot_diff
OUTPUTDIR=${MENUPATH}/output_files
TEMP_CLEAN=/tmp/clean
TEMP_CHANGED=/tmp/changed
DATAFILE1=${MENUPATH}/raw_files_path
DATAFILE_CLEAN=${MENUPATH}/files_list_clean
DATAFILE_CHANGED=${MENUPATH}/files_list_changed
OUTPUT_ADDED=${OUTPUTDIR}/files_added.list
OUTPUT_DELETED=${OUTPUTDIR}/files_deleted.list
OUTPUT_EDITED=${OUTPUTDIR}/files_edited.list
OUTPUT_SETID=${OUTPUTDIR}/files_setid.list

lne=5
cl=5
CTRAP
tput clear
tput smso
tput cup $lne $cl
echo -n "VM SNAPSHOT ANALYSIS"
lne=`expr $lne + 2`
tput cup $lne $cl
echo -n "View Select Snapshots"
tput rmso
if [ -f $DATAFILE1 ]; then
    file1=`sed -n 1p $DATAFILE1`
    file2=`sed -n 2p $DATAFILE1`
    if [ ! -z $file1 ]&&[ ! -z $file2 ]; then
        lne=`expr $lne + 2`
        tput cup $lne $cl
    fi
fi
```

```
        echo -n "Baseline snapshot: $file1"
        lne=`expr $lne + 2`
        tput cup $lne $cl
        echo -n "Suspected snapshot: $file2"
        lne=`expr $lne + 2`
        tput cup $lne $cl
        echo -n "Time Skew: $skew seconds"
    else
        NO_FILE
    fi
else
    NO_FILE
fi
lne=`expr $lne + 2`
tput cup $lne $cl
echo -n "Please press <ENTER> to continue."
read
```

**deleted\_files.sh**

```
#!/bin/bash

function CTRAP {
    trap '
        let lne++
        let lne++
        tput cup $lne $cl
        echo -n "CTRL_C Not Allowed."
        let lne++
        tput cup $lne $cl
        echo -n "Please press <ENTER> to continue."
    ' 2
}

function PROG_BAR {
    while [ `ps ax | grep $pid | grep -v grep | awk '{ print $1
}'` ]
    do
        tput cup $lne 5
        echo -n "|"
        tput cup $lne 57
        echo -n "|"
        tput cup $lne 6
        for (( i = 1 ; i < 51 ; i++ ))
        do
            echo -n " "
            usleep $scrttime
        done
    done
}

MENUPATH=/home/manish/snapshot_diff
OUTPUTDIR=${MENUPATH}/output_files
TEMP_CLEAN=/tmp/clean
TEMP_CHANGED=/tmp/changed
DATAFILE1=${MENUPATH}/raw_files_path
DATAFILE_CLEAN=${MENUPATH}/files_list_clean
DATAFILE_CHANGED=${MENUPATH}/files_list_changed
OUTPUT_ADDED=${OUTPUTDIR}/files_added.list
OUTPUT_DELETED=${OUTPUTDIR}/files_deleted.list
OUTPUT_EDITED=${OUTPUTDIR}/files_edited.list
OUTPUT_SETID=${OUTPUTDIR}/files_setid.list

lne=5
cl=5
CTRAP
tput clear
tput smso
```



```

tput cup $lne $cl
echo -n "VM SNAPSHOT ANALYSIS"
lne=`expr $lne + 2`
tput cup $lne $cl
echo -n "View Files Deleted"
tput rmso
lne=`expr $lne + 2`
tput cup $lne $cl
echo -n "The disk partitons will be displayed"
let lne++
tput cup $lne $cl
echo -n "for selection and analysis."
lne=`expr $lne + 2`
tput cup $lne $cl
echo -n "Please press <ENTER> to continue. To EXIT press any
other key."
read -n1 action
if [ -z $action ]; then
    tput clear
    lne=5
    cl=5
    tput clear
    tput smso
    tput cup $lne $cl
    echo -n "VM SNAPSHOT ANALYSIS"
    lne=`expr $lne + 2`
    tput cup $lne $cl
    echo -n "View Files Deleted"
    tput rmso
    lne=`expr $lne + 2`
    tput cup $lne $cl
    echo -n "The partition are listed below:"
    lne=`expr $lne + 2`
    tput cup $lne 0
    file1=`sed -n 1p $DATAFILE1`
#    file2=` echo $file1 | awk -F'/' '{ print $NF }'` # Get last
segment
#    file3=`echo $file1 | sed s/$file2//`
    cd ${file1%/*} && /sbin/fdisk -l ${file1##*/} | grep -iv
swap
    cd $MENUPATH
## Get cursor positon
    function GET_CURSOR {
        stty -echo
        echo -n $'\e[6n'
        read -d R x; stty echo
        lne=`echo ${x#??} | cut -f1 -d';'`
        lne=`expr $lne - 2`
    }
    GET_CURSOR
## Get cursor positon
    function SECT {

```

```

        line=`expr $line + 2`
        tput cup $line $cl
        echo -n "Please enter the start of the partiton sector
to analyze: "

        read sector
        if [ -z $sector ]; then
            let line++
            tput cup $line $cl
            echo -n "Input cannot be left blank."
            let line++
            tput cup $line $cl
            echo -n "Do you wish to continue (y/n): "
            let line++
            tput cup $line $cl
            echo -n "Enter N/n to exit."
            line=`expr $line - 1`
            tput cup $line 36
            read ans
            if [ $ans == n -o $ans == N ] &> /dev/null ; then
                exit
            else
                SECT
            fi
        else
            if [[ $sector == *([0-9]) ]]; then
                file1=`sed -n 1p $DATAFILE1`
                file2=`sed -n 2p $DATAFILE1`
                /usr/bin/fls -m / -r -f ext -i raw -o
$sector $file1 | cut -f 2 -d'|' > $DATAFILE_CLEAN &
                pid=`pgrep fls | head -1`
                # pid=`echo $!`
                let line++
                tput cup $line $cl
                echo -n "Content of clean snapshot are
being recorded, please wait:"
                let line++
                scrttime=200000
                PROG_BAR
                /usr/bin/fls -m / -r -f ext -i raw -o
$sector $file2 | cut -f 2 -d'|' > $DATAFILE_CHANGED &
                pid=`pgrep fls | head -1`
                # pid=`echo $!`
                line=`expr $line + 2`
                tput cup $line $cl
                echo -n "Content of suspect snapshot are
being recorded, please wait:"
                let line++
                scrttime=200000
                PROG_BAR
                line=`expr $line + 2`
                tput cup $line $cl
                echo -n "Contents of snapshots gathered."

```

```

lne=`expr $lne + 2`
tput cup $lne $cl
echo -n "To proceed for Analysis please
press <ENTER>."

read
tput clear
lne=5
cl=5
tput clear
tput smso
tput cup $lne $cl
echo -n "VM SNAPSHOT ANALYSIS"
lne=`expr $lne + 2`
tput cup $lne $cl
echo -n "View Files Deleted"
tput rmso
lne=`expr $lne + 2`
tput cup $lne $cl
echo -n "Analysing files, please wait.."
echo "" > $OUTPUT_DELETED
echo "" >> $OUTPUT_DELETED
echo "      Following are the files that
have been deleted:" >> $OUTPUT_DELETED
echo "" >> $OUTPUT_DELETED
echo "" >> $OUTPUT_DELETED
lne=`expr $lne + 2`
tput cup $lne $cl
/bin/sort $DATAFILE_CLEAN > $TEMP_CLEAN &
pid=`echo $!`
sctrtime=50000
PROG_BAR
/bin/sort $DATAFILE_CHANGED > $TEMP_CHANGED
&

pid=`echo $!`
sctrtime=50000
PROG_BAR
/usr/bin/diff $TEMP_CLEAN $TEMP_CHANGED |
grep \< | sed s/\<\ // >> $OUTPUT_DELETED &
pid=`echo $!`
sctrtime=50000
PROG_BAR
lne=`expr $lne + 2`
tput cup $lne $cl
echo -n "A list of files deleted from the
VM has be"

let lne++
tput cup $lne $cl
echo -n "populated and stored at:
$OUTPUT_DELETED"

let lne++
tput cup $lne $cl

```

```

file."
                                echo -n "Please press <ENTER> to view the
                                read
                                line_infile=`wc -l $OUTPUT_DELETED | awk '{
print $1 }'`
                                if [ $line_infile -gt 5 ]; then
                                tput clear
                                tput cup 3 5
                                echo -n "Following are the files that
                                #
                                #
                                have been deleted:"
                                #
                                tput cup 5 0
                                /bin/more $OUTPUT_DELETED
                                echo ; echo
                                GET_CURSOR
                                else
                                lne=`expr $lne + 2`
                                tput cup $lne $cl
                                tput smso
                                echo -n "No files have been deleted.

                                tput rmso
                                fi
                                lne=`expr $lne + 2`
                                tput cup $lne $cl
                                echo -n "Please press <ENTER> to continue."
                                read
                                else
                                lne=`expr $lne + 2`
                                tput cup $lne $cl
                                echo "The input is not a number."
                                let lne++
                                tput cup $lne $cl
                                echo -n "Do you wish to continue (y/n): "
                                let lne++
                                tput cup $lne $cl
                                echo -n "Enter N/n to exit."
                                line=`expr $lne - 1`
                                tput cup $line 36
                                read ans
                                if [ $ans == n -o $ans == N ] &> /dev/null
                                ; then
                                exit
                                else
                                SECT
                                fi
                                fi
                                fi
                                }
                                SECT
                                fi

```

**added\_files.sh**

```
#!/bin/bash

function CTRAP {
    trap '
        let lne++
        let lne++
        tput cup $lne $cl
        echo -n "CTRL_C Not Allowed."
        let lne++
        tput cup $lne $cl
        echo -n "Please press <ENTER> to continue."
    ' 2
}

function PROG_BAR {
    while [ `ps ax | grep $pid | grep -v grep | awk '{ print $1
}'` ]
    do
        tput cup $lne 5
        echo -n "|"
        tput cup $lne 57
        echo -n "|"
        tput cup $lne 6
        for (( i = 1 ; i < 51 ; i++ ))
        do
            echo -n " "
            usleep $scrttime
        done
    done
}

MENUPATH=/home/manish/snapshot_diff
OUTPUTDIR=${MENUPATH}/output_files
TEMP_CLEAN=/tmp/clean
TEMP_CHANGED=/tmp/changed
DATAFILE1=${MENUPATH}/raw_files_path
DATAFILE_CLEAN=${MENUPATH}/files_list_clean
DATAFILE_CHANGED=${MENUPATH}/files_list_changed
OUTPUT_ADDED=${OUTPUTDIR}/files_added.list
OUTPUT_DELETED=${OUTPUTDIR}/files_deleted.list
OUTPUT_EDITED=${OUTPUTDIR}/files_edited.list
OUTPUT_SETID=${OUTPUTDIR}/files_setid.list

lne=5
cl=5
CTRAP
tput clear
tput smso
tput cup $lne $cl
echo -n "VM SNAPSHOT ANALYSIS"
```

```

lne=`expr $lne + 2`
tput cup $lne $cl
echo -n "View Files Added to New Snapshot"
tput rmso
lne=`expr $lne + 2`
tput cup $lne $cl
echo -n "The disk partitons will be displayed"
let lne++
tput cup $lne $cl
echo -n "for selection and analysis."
lne=`expr $lne + 2`
tput cup $lne $cl
echo -n "Please press <ENTER> to continue. To EXIT press any
other key."
read -n1 action
if [ -z $action ]; then
    tput clear
    lne=5
    cl=5
    tput clear
    tput smso
    tput cup $lne $cl
    echo -n "VM SNAPSHOT ANALYSIS"
    lne=`expr $lne + 2`
    tput cup $lne $cl
    echo -n "View Files Added to New Snapshot"
    tput rmso
    lne=`expr $lne + 2`
    tput cup $lne $cl
    echo -n "The partition are listed below:"
    lne=`expr $lne + 2`
    tput cup $lne 0
    file1=`sed -n 1p $DATAFILE1`
#    file2=` echo $file1 | awk -F '/' '{ print $NF }'` # Get last
segment
#    file3=`echo $file1 | sed s/$file2//`
    cd ${file1%/*} && /sbin/fdisk -l ${file1##*/} | grep -iv
swap
    cd $MENUPATH
## Get cursor positon
function GET_CURSOR {
    stty -echo
    echo -n $'\e[6n'
    read -d R x; stty echo
    lne=`echo ${x#??} | cut -f1 -d';'`
    lne=`expr $lne - 2`
}
GET_CURSOR
## Get cursor positon
function SECT {
    lne=`expr $lne + 2`
    tput cup $lne $cl

```

```

to analyze: "
    echo -n "Please enter the start of the partiton sector
    read sector
    if [ -z $sector ]; then
        let lne++
        tput cup $lne $cl
        echo -n "Input cannot be left blank."
        let lne++
        tput cup $lne $cl
        echo -n "Do you wish to continue (y/n): "
        let lne++
        tput cup $lne $cl
        echo -n "Enter N/n to exit."
        line=`expr $lne - 1`
        tput cup $line 36
        read ans
        if [ $ans == n -o $ans == N ] &> /dev/null ; then
            exit
        else
            SECT
        fi
    else
        if [[ $sector == *([0-9]) ]]; then
            file1=`sed -n 1p $DATAFILE1`
            file2=`sed -n 2p $DATAFILE1`
            /usr/bin/fls -m / -r -f ext -i raw -o
$sector $file1 | cut -f 2 -d'|' > $DATAFILE_CLEAN &
            pid=`pgrep fls | head -1`
#            pid=`echo $!`
            let lne++
            tput cup $lne $cl
            echo -n "Content of clean snapshot are
being recorded, please wait:"
            let lne++
            scrttime=200000
            PROG_BAR
            /usr/bin/fls -m / -r -f ext -i raw -o
$sector $file2 | cut -f 2 -d'|' > $DATAFILE_CHANGED &
            pid=`pgrep fls | head -1`
#            pid=`echo $!`
            lne=`expr $lne + 2`
            tput cup $lne $cl
            echo -n "Content of suspect snapshot are
being recorded, please wait:"
            let lne++
            scrttime=200000
            PROG_BAR
            lne=`expr $lne + 2`
            tput cup $lne $cl
            echo -n "Contents of snapshots gathered."
            lne=`expr $lne + 2`
            tput cup $lne $cl

```

```

echo -n "To proceed for Analysis please
press <ENTER>."

read
tput clear
lne=5
cl=5
tput clear
tput smso
tput cup $lne $cl
echo -n "VM SNAPSHOT ANALYSIS"
lne=`expr $lne + 2`
tput cup $lne $cl
echo -n "View Files Added to New Snapshot"
tput rmso
lne=`expr $lne + 2`
tput cup $lne $cl
echo -n "Analysing files, please wait.."
echo "" > $OUTPUT_ADDED
echo "" >> $OUTPUT_ADDED
echo "    Following are the new files

added:" >> $OUTPUT_ADDED

echo "" >> $OUTPUT_ADDED
echo "" >> $OUTPUT_ADDED
lne=`expr $lne + 2`
tput cup $lne $cl
/bin/sort $DATAFILE_CLEAN > $TEMP_CLEAN &
pid=`echo $!`
sctrtime=50000
PROG_BAR
/bin/sort $DATAFILE_CHANGED > $TEMP_CHANGED

&

pid=`echo $!`
sctrtime=50000
PROG_BAR
/usr/bin/diff $TEMP_CLEAN $TEMP_CHANGED |
grep \> | sed s/\>\ // >> $OUTPUT_ADDED &
pid=`echo $!`
sctrtime=50000
PROG_BAR
lne=`expr $lne + 2`
tput cup $lne $cl
echo -n "A list of new files added to the

VM has be"

let lne++
tput cup $lne $cl
echo -n "populated and stored at:

$OUTPUT_ADDED"

let lne++
tput cup $lne $cl
echo -n "Please press <ENTER> to view the

file."

read

```



```

print $1 }''

#
#
added:"
#

line_infile=`wc -l $OUTPUT_ADDED | awk '{
if [ $line_infile -gt 5 ]; then
    tput clear
    tput cup 3 5
    echo -n "Following are the new files

    tput cup 5 0
    /bin/more $OUTPUT_ADDED
    echo ; echo
    GET_CURSOR
else
    lne=`expr $lne + 2`
    tput cup $lne $cl
    tput smso
    echo -n "No files have been added.

    tput rmso
fi
lne=`expr $lne + 2`
tput cup $lne $cl
echo -n "Please press <ENTER> to continue."
read
else
    lne=`expr $lne + 2`
    tput cup $lne $cl
    echo "The input is not a number."
    let lne++
    tput cup $lne $cl
    echo -n "Do you wish to continue (y/n): "
    let lne++
    tput cup $lne $cl
    echo -n "Enter N/n to exit."
    line=`expr $lne - 1`
    tput cup $line 36
    read ans
    if [ $ans == n -o $ans == N ] &> /dev/null

; then

        exit
    else
        SECT
    fi
fi
fi
}
SECT
fi

```

**edited\_files.sh**

```
#!/bin/bash

function CTRAP {
    trap '
        let lne++
        let lne++
        tput cup $lne $cl
        echo -n "CTRL_C Not Allowed."
        let lne++
        tput cup $lne $cl
        echo -n "Please press <ENTER> to continue."
    ' 2
}

function PROG_BAR {
    while [ `ps ax | grep $pid | grep -v grep | awk '{ print $1
}'` ]
        do
            tput cup $lne 5
            echo -n "|"
            tput cup $lne 57
            echo -n "|"
            tput cup $lne 6
            for (( i = 1 ; i < 51 ; i++ ))
            do
                echo -n " "
                usleep $scrttime
            done
        done
}

MENUPATH=/home/manish/snapshot_diff
OUTPUTDIR=${MENUPATH}/output_files
TEMP_CLEAN=/tmp/clean
TEMP_CHANGED=/tmp/changed
DATAFILE1=${MENUPATH}/raw_files_path
DATAFILE_CLEAN=${MENUPATH}/files_list_clean
DATAFILE_CHANGED=${MENUPATH}/files_list_changed
OUTPUT_ADDED=${OUTPUTDIR}/files_added.list
OUTPUT_DELETED=${OUTPUTDIR}/files_deleted.list
OUTPUT_EDITED=${OUTPUTDIR}/files_edited.list
OUTPUT_SETID=${OUTPUTDIR}/files_setid.list

lne=5
cl=5
CTRAP
tput clear
tput smso
```

```

tput cup $lne $cl
echo -n "VM SNAPSHOT ANALYSIS"
lne=`expr $lne + 2`
tput cup $lne $cl
echo -n "View Files Edited [Modification Time]"
tput rmso
lne=`expr $lne + 2`
tput cup $lne $cl
echo -n "The disk partitons will be displayed"
let lne++
tput cup $lne $cl
echo -n "for selection and analysis."
lne=`expr $lne + 2`
tput cup $lne $cl
echo -n "Please press <ENTER> to continue. To EXIT press any
other key."
read -n1 action
if [ -z $action ]; then
    tput clear
    lne=5
    cl=5
    tput clear
    tput smso
    tput cup $lne $cl
    echo -n "VM SNAPSHOT ANALYSIS"
    lne=`expr $lne + 2`
    tput cup $lne $cl
    echo -n "View Files Edited [Modification Time]"
    tput rmso
    lne=`expr $lne + 2`
    tput cup $lne $cl
    echo -n "The partition are listed below:"
    lne=`expr $lne + 2`
    tput cup $lne 0
    file1=`sed -n 1p $DATAFILE1`
#    file2=` echo $file1 | awk -F'/' '{ print $NF }'` # Get last
segment
#    file3=`echo $file1 | sed s/$file2//`
    cd ${file1%/*} && /sbin/fdisk -l ${file1##*/} | grep -iv
swap
    cd $MENUPATH
## Get cursor positon
    function GET_CURSOR {
        stty -echo
        echo -n $'\e[6n'
        read -d R x; stty echo
        lne=`echo ${x#??} | cut -f1 -d';'`
        lne=`expr $lne - 2`
    }
    GET_CURSOR
## Get cursor positon
    function SECT {

```

```

        lne=`expr $lne + 2`
        tput cup $lne $cl
        echo -n "Please enter the start of the partiton sector
to analyze: "
        read sector
        if [ -z $sector ]; then
            let lne++
            tput cup $lne $cl
            echo -n "Input cannot be left blank."
            let lne++
            tput cup $lne $cl
            echo -n "Do you wish to continue (y/n): "
            let lne++
            tput cup $lne $cl
            echo -n "Enter N/n to exit."
            line=`expr $lne - 1`
            tput cup $line 36
            read ans
            if [ $ans == n -o $ans == N ] &> /dev/null ; then
                exit
            else
                SECT
            fi
        else
            if [[ $sector == *([0-9]) ]]; then
                file1=`sed -n 1p $DATAFILE1`
                file2=`sed -n 2p $DATAFILE1`
                skew=`sed -n 3p $DATAFILE1`
                /usr/bin/fls -m / -r -f ext -i raw -s $skew
-o $sector $file1 | cut -f 2,9 -d'|' > $DATAFILE_CLEAN &
                pid=`pgrep fls | head -1`
                # pid=`echo $!`
                let lne++
                tput cup $lne $cl
                echo -n "Content of clean snapshot are
being recorded, please wait:"
                let lne++
                scrttime=200000
                PROG_BAR
                /usr/bin/fls -m / -r -f ext -i raw -s $skew
-o $sector $file2 | cut -f 2,9 -d'|' > $DATAFILE_CHANGED &
                pid=`pgrep fls | head -1`
                # pid=`echo $!`
                lne=`expr $lne + 2`
                tput cup $lne $cl
                echo -n "Content of suspect snapshot are
being recorded, please wait:"
                let lne++
                scrttime=200000
                PROG_BAR
                lne=`expr $lne + 2`
                tput cup $lne $cl

```

```

echo -n "Contents of snapshots gathered."
lne=`expr $lne + 2`
tput cup $lne $cl
echo -n "To proceed for Analysis please
press <ENTER>."

read
tput clear
lne=5
cl=5
tput clear
tput smso
tput cup $lne $cl
echo -n "VM SNAPSHOT ANALYSIS"
lne=`expr $lne + 2`
tput cup $lne $cl
echo -n "View Files Edited [Modification
Time]"

tput rmso
lne=`expr $lne + 2`
tput cup $lne $cl
echo -n "Analysing files, please wait.."
lne=`expr $lne + 2`
tput cup $lne $cl
echo -n "This is a very time intensive
process and might take"

let lne++
tput cup $lne $cl
echo -n "a several hours, please leave the
program running."

let lne++
tput cup $lne $cl
echo -n "No progress bar will be
displayed."

echo "" > $OUTPUT_EDITED
echo "" >> $OUTPUT_EDITED
echo "    Following are the files that
have been edited:" >> $OUTPUT_EDITED
echo "" >> $OUTPUT_EDITED
echo "" >> $OUTPUT_EDITED
/bin/sort $DATAFILE_CLEAN > $TEMP_CLEAN
/bin/sort $DATAFILE_CHANGED > $TEMP_CHANGED
#
cat /dev/null > $OUTPUT_EDITED
num_lines=`wc -l $TEMP_CLEAN | awk '{ print
$1 }'`

for (( i = 1 ; i <= $num_lines ; i++ ))
do
    line_file=`sed -n ${i}p $TEMP_CLEAN |
sed 's/\[/\\\[/'`

    m_time_clean=`echo ${line_file###|}`
    if [ $m_time_clean -ne 0 ] ; then
        grep_word=`echo $line_file | sed
s/$m_time_clean//`

```

```

result=`grep "^$grep_word"
$TEMP_CHANGED`
if [ -z $result ] &> /dev/null ;
then
:
else
m_time_changed=`echo
${result##*|}`
if [ $m_time_changed -ne 0
] ; then
if [ $m_time_changed
-ne $m_time_clean ] ; then
# echo "`date --
date="1970-01-01 00:00:00 UTC +${m_time_clean} seconds" +%c`
${line_file%|*} `date --date="1970-01-01 00:00:00 UTC
+${m_time_changed} seconds" +%c`" >> $OUTPUT_EDITED
echo -e
"Original Modification Time\t`date --date="1970-01-01 00:00:00 UTC
+${m_time_clean} seconds" +%c`\nFile
name\t\t\t\t${line_file%|*}\nCurrent Modification Time\t`date --
date="1970-01-01 00:00:00 UTC +${m_time_changed} seconds" +%c`\n" >>
$OUTPUT_EDITED
fi
fi
fi
done
lne=`expr $lne + 2`
tput cup $lne $cl
echo -n "A list of files edited in the VM
has be"
let lne++
tput cup $lne $cl
echo -n "populated and stored at:
$OUTPUT_EDITED"
let lne++
tput cup $lne $cl
echo -n "Please press <ENTER> to view the
file."
read
line_infile=`wc -l $OUTPUT_EDITED | awk '{
print $1 }'`
if [ $line_infile -gt 5 ]; then
tput clear
tput cup 3 5
echo -n "Following are the files that
#
#
have been edited:"
#
tput cup 5 0
/bin/more $OUTPUT_EDITED
echo ; echo
GET_CURSOR
else

```

```

        lne=`expr $lne + 2`
        tput cup $lne $cl
        tput smso
        echo -n "No files have been edited!!"
        let lne++
        tput cup $lne $cl
        tput smso
        echo -n "Are you sure they are two
different snapshots?"
        tput rmso
    fi
    lne=`expr $lne + 2`
    tput cup $lne $cl
    echo -n "Please press <ENTER> to continue."
    read
else
    lne=`expr $lne + 2`
    tput cup $lne $cl
    echo "The input is not a number."
    let lne++
    tput cup $lne $cl
    echo -n "Do you wish to continue (y/n): "
    let lne++
    tput cup $lne $cl
    echo -n "Enter N/n to exit."
    line=`expr $lne - 1`
    tput cup $line 36
    read ans
    if [ $ans == n -o $ans == N ] &> /dev/null
; then
        exit
    else
        SECT
    fi
fi
fi
}
SECT
fi

```

**change\_files.sh**

```
#!/bin/bash

function CTRAP {
    trap '
        let lne++
        let lne++
        tput cup $lne $cl
        echo -n "CTRL_C Not Allowed."
        let lne++
        tput cup $lne $cl
        echo -n "Please press <ENTER> to continue."
    ' 2
}

function PROG_BAR {
    while [ `ps ax | grep $pid | grep -v grep | awk '{ print $1
}'` ]
    do
        tput cup $lne 5
        echo -n "|"
        tput cup $lne 57
        echo -n "|"
        tput cup $lne 6
        for (( i = 1 ; i < 51 ; i++ ))
        do
            echo -n " "
            usleep $scrttime
        done
    done
}

MENUPATH=/home/manish/snapshot_diff
OUTPUTDIR=${MENUPATH}/output_files
TEMP_CLEAN=/tmp/clean
TEMP_CHANGED=/tmp/changed
DATAFILE1=${MENUPATH}/raw_files_path
DATAFILE_CLEAN=${MENUPATH}/files_list_clean
DATAFILE_CHANGED=${MENUPATH}/files_list_changed
OUTPUT_ADDED=${OUTPUTDIR}/files_added.list
OUTPUT_DELETED=${OUTPUTDIR}/files_deleted.list
OUTPUT_EDITED=${OUTPUTDIR}/files_edited.list
OUTPUT_CHANGED=${OUTPUTDIR}/files_changed.list
OUTPUT_SETID=${OUTPUTDIR}/files_setid.list

lne=5
cl=5
CTRAP
tput clear
```



```

tput smso
tput cup $lne $cl
echo -n "VM SNAPSHOT ANALYSIS"
lne=`expr $lne + 2`
tput cup $lne $cl
echo -n "View Files Changed [Change Time]"
tput rmso
lne=`expr $lne + 2`
tput cup $lne $cl
echo -n "The disk partitons will be displayed"
let lne++
tput cup $lne $cl
echo -n "for selection and analysis."
lne=`expr $lne + 2`
tput cup $lne $cl
echo -n "Please press <ENTER> to continue. To EXIT press any
other key."
read -n1 action
if [ -z $action ]; then
    tput clear
    lne=5
    cl=5
    tput clear
    tput smso
    tput cup $lne $cl
    echo -n "VM SNAPSHOT ANALYSIS"
    lne=`expr $lne + 2`
    tput cup $lne $cl
    echo -n "View Files Changed [Change Time]"
    tput rmso
    lne=`expr $lne + 2`
    tput cup $lne $cl
    echo -n "The partition are listed below:"
    lne=`expr $lne + 2`
    tput cup $lne 0
    file1=`sed -n 1p $DATAFILE1`
#    file2=`echo $file1 | awk -F'/' '{ print $NF }'` # Get last
segment
#    file3=`echo $file1 | sed s/$file2//`
    cd ${file1%/*} && /sbin/fdisk -l ${file1##*/} | grep -iv
swap
    cd $MENUPATH
## Get cursor positon
    function GET_CURSOR {
        stty -echo
        echo -n $'\e[6n'
        read -d R x; stty echo
        lne=`echo ${x#??} | cut -f1 -d';'`
        lne=`expr $lne - 2`
    }
    GET_CURSOR
## Get cursor positon

```

```

function SECT {
    lne=`expr $lne + 2`
    tput cup $lne $cl
    echo -n "Please enter the start of the partiton sector
to analyze: "
    read sector
    if [ -z $sector ]; then
        let lne++
        tput cup $lne $cl
        echo -n "Input cannot be left blank."
        let lne++
        tput cup $lne $cl
        echo -n "Do you wish to continue (y/n): "
        let lne++
        tput cup $lne $cl
        echo -n "Enter N/n to exit."
        line=`expr $lne - 1`
        tput cup $line 36
        read ans
        if [ $ans == n -o $ans == N ] &> /dev/null ; then
            exit
        else
            SECT
        fi
    else
        if [[ $sector == *([0-9]) ]]; then
            file1=`sed -n 1p $DATAFILE1`
            file2=`sed -n 2p $DATAFILE1`
            skew=`sed -n 3p $DATAFILE1`
            /usr/bin/fls -m / -r -f ext -i raw -s $skew
            -o $sector $file1 | cut -f 2,10 -d'|' > $DATAFILE_CLEAN &
            pid=`pgrep fls | head -1`
            # pid=`echo $!`
            let lne++
            tput cup $lne $cl
            echo -n "Content of clean snapshot are
being recorded, please wait:"
            let lne++
            scrttime=200000
            PROG_BAR
            /usr/bin/fls -m / -r -f ext -i raw -s $skew
            -o $sector $file2 | cut -f 2,10 -d'|' > $DATAFILE_CHANGED &
            pid=`pgrep fls | head -1`
            # pid=`echo $!`
            lne=`expr $lne + 2`
            tput cup $lne $cl
            echo -n "Content of suspect snapshot are
being recorded, please wait:"
            let lne++
            scrttime=200000
            PROG_BAR
            lne=`expr $lne + 2`

```

```

tput cup $line $cl
echo -n "Contents of snapshots gathered."
line=`expr $line + 2`
tput cup $line $cl
echo -n "To proceed for Analysis please
press <ENTER>."

read
tput clear
line=5
cl=5
tput clear
tput smso
tput cup $line $cl
echo -n "VM SNAPSHOT ANALYSIS"
line=`expr $line + 2`
tput cup $line $cl
echo -n "View Files Changed [Change Time]"
tput rmso
line=`expr $line + 2`
tput cup $line $cl
echo -n "Analysing files, please wait.."
line=`expr $line + 2`
tput cup $line $cl
echo -n "This is a very time intensive
process and might take"

let line++
tput cup $line $cl
echo -n "a several hours, please leave the
program running."

let line++
tput cup $line $cl
echo -n "No progress bar will be
displayed."

echo "" > $OUTPUT_CHANGED
echo "" >> $OUTPUT_CHANGED
echo "      Following are the files that
have been changed:" >> $OUTPUT_CHANGED
echo "" >> $OUTPUT_CHANGED
echo "" >> $OUTPUT_CHANGED
/bin/sort $DATAFILE_CLEAN > $TEMP_CLEAN
/bin/sort $DATAFILE_CHANGED > $TEMP_CHANGED
cat /dev/null > $OUTPUT_CHANGED
num_lines=`wc -l $TEMP_CLEAN | awk '{ print
$1 }'`

for (( i = 1 ; i <= $num_lines ; i++ ))
do
    line_file=`sed -n ${i}p $TEMP_CLEAN |
sed 's/\[/\\[/\'`

    c_time_clean=`echo ${line_file##*|}}`
    if [ $c_time_clean -ne 0 ] ; then
        grep_word=`echo $line_file | sed
s/$c time clean/`

```

```

result=`grep "^$grep_word"
$TEMP_CHANGED`
if [ -z $result ] &> /dev/null ;
then
:
else
c_time_changed=`echo
${result##*|}`
if [ $c_time_changed -ne 0
] ; then
if [ $c_time_changed
-ne $c_time_clean ] ; then
# echo "`date --
date="1970-01-01 00:00:00 UTC +${c_time_clean} seconds" +%c`
${line_file%|*} `date --date="1970-01-01 00:00:00 UTC
+${c_time_changed} seconds" +%c`" >> $OUTPUT_CHANGED
echo -e
"Original Change Time\t`date --date="1970-01-01 00:00:00 UTC
+${c_time_clean} seconds" +%c`\nFile name\t\t${line_file%|*}\nCurrent
Change Time\t`date --date="1970-01-01 00:00:00 UTC +${c_time_changed}
seconds" +%c`\n" >> $OUTPUT_CHANGED
fi
fi
fi
done
lne=`expr $lne + 2`
tput cup $lne $cl
echo -n "A list of files changed in the VM
has be"
let lne++
tput cup $lne $cl
echo -n "populated and stored at:
$OUTPUT_CHANGED"
let lne++
tput cup $lne $cl
echo -n "Please press <ENTER> to view the
file."
read
line_infile=`wc -l $OUTPUT_CHANGED | awk '{
print $1 }'`
if [ $line_infile -gt 5 ]; then
tput clear
tput cup 3 5
echo -n "Following are the files that
#
#
have been changed:"
#
tput cup 5 0
/bin/more $OUTPUT_CHANGED
echo ; echo
GET_CURSOR
else
lne=`expr $lne + 2`

```

```

        tput cup $lne $cl
        tput smso
        echo -n "No files have been changed!!"
        let lne++
        tput cup $lne $cl
        tput smso
        echo -n "Are you sure they are two
different snapshots?"

        tput rmso
    fi
    lne=`expr $lne + 2`
    tput cup $lne $cl
    echo -n "Please press <ENTER> to continue."
    read
else
    lne=`expr $lne + 2`
    tput cup $lne $cl
    echo "The input is not a number."
    let lne++
    tput cup $lne $cl
    echo -n "Do you wish to continue (y/n): "
    let lne++
    tput cup $lne $cl
    echo -n "Enter N/n to exit."
    line=`expr $lne - 1`
    tput cup $line 36
    read ans
    if [ $ans == n -o $ans == N ] &> /dev/null
; then
        exit
    else
        SECT
    fi
fi
fi
}
SECT
fi

```

**setid\_changes.sh**

```
#!/bin/bash

function CTRAP {
    trap '
        let lne++
        let lne++
        tput cup $lne $cl
        echo -n "CTRL_C Not Allowed."
        let lne++
        tput cup $lne $cl
        echo -n "Please press <ENTER> to continue."
    ' 2
}

function PROG_BAR {
    while [ `ps ax | grep $pid | grep -v grep | awk '{ print $1
}'` ]
    do
        tput cup $lne 5
        echo -n "|"
        tput cup $lne 57
        echo -n "|"
        tput cup $lne 6
        for (( i = 1 ; i < 51 ; i++ ))
        do
            echo -n " "
            usleep $scrttime
        done
    done
}

MENUPATH=/home/manish/snapshot_diff
OUTPUTDIR=${MENUPATH}/output_files
TEMP_CLEAN=/tmp/clean
TEMP_CHANGED=/tmp/changed
DATAFILE1=${MENUPATH}/raw_files_path
DATAFILE_CLEAN=${MENUPATH}/files_list_clean
DATAFILE_CHANGED=${MENUPATH}/files_list_changed
OUTPUT_ADDED=${OUTPUTDIR}/files_added.list
OUTPUT_DELETED=${OUTPUTDIR}/files_deleted.list
OUTPUT_EDITED=${OUTPUTDIR}/files_edited.list
OUTPUT_SETID=${OUTPUTDIR}/files_setid.list

lne=5
cl=5
CTRAP
tput clear
tput smso
tput cup $lne $cl
```

```

echo -n "VM SNAPSHOT ANALYSIS"
lne=`expr $lne + 2`
tput cup $lne $cl
echo -n "View Files with Recent SETUID/SETGID Changes"
tput rmso
lne=`expr $lne + 2`
tput cup $lne $cl
echo -n "The disk partitons will be displayed"
let lne++
tput cup $lne $cl
echo -n "for selection and analysis."
lne=`expr $lne + 2`
tput cup $lne $cl
echo -n "Please press <ENTER> to continue. To EXIT press any
other key."
read -n1 action
if [ -z $action ]; then
    tput clear
    lne=5
    cl=5
    tput clear
    tput smso
    tput cup $lne $cl
    echo -n "VM SNAPSHOT ANALYSIS"
    lne=`expr $lne + 2`
    tput cup $lne $cl
    echo -n "View Files with Recent SETUID/SETGID Changes"
    tput rmso
    lne=`expr $lne + 2`
    tput cup $lne $cl
    echo -n "The partition are listed below:"
    lne=`expr $lne + 2`
    tput cup $lne 0
    file1=`sed -n 1p $DATAFILE1`
#    file2=` echo $file1 | awk -F'/' '{ print $NF }'` # Get last
segment
#    file3=`echo $file1 | sed s/$file2//`
    cd ${file1%/*} && /sbin/fdisk -l ${file1##*/} | grep -iv
swap
    cd $MENUPATH
## Get cursor positon
    function GET_CURSOR {
        stty -echo
        echo -n $'\e[6n'
        read -d R x; stty echo
        lne=`echo ${x#??} | cut -f1 -d';'`
        lne=`expr $lne - 2`
    }
    GET_CURSOR
## Get cursor positon
    function SECT {
        lne=`expr $lne + 2`

```

```

tput cup $lne $cl
echo -n "Please enter the start of the partiton sector
to analyze: "
read sector
if [ -z $sector ]; then
    let lne++
    tput cup $lne $cl
    echo -n "Input cannot be left blank."
    let lne++
    tput cup $lne $cl
    echo -n "Do you wish to continue (y/n): "
    let lne++
    tput cup $lne $cl
    echo -n "Enter N/n to exit."
    line=`expr $lne - 1`
    tput cup $line 36
    read ans
    if [ $ans == n -o $ans == N ] &> /dev/null ; then
        exit
    else
        SECT
    fi
else
    if [[ $sector == *([0-9]) ]]; then
        file1=`sed -n 1p $DATAFILE1`
        file2=`sed -n 2p $DATAFILE1`
        /usr/bin/fls -m / -r -f ext -i raw -o
$sector $file1 | cut -f 2,4 -d'|' | grep -iE "\|...r.s\|.....r.s" >
$DATAFILE_CLEAN &

        pid=`pgrep fls | head -1`
        pid=`echo $!`
        let lne++
        tput cup $lne $cl
        echo -n "Content of clean snapshot are
being recorded, please wait:"
        let lne++
        scrttime=200000
        PROG_BAR
        /usr/bin/fls -m / -r -f ext -i raw -o
$sector $file2 | cut -f 2,4 -d'|' | grep -iE "\|...r.s\|.....r.s" >
$DATAFILE_CHANGED &

        pid=`pgrep fls | head -1`
        pid=`echo $!`
        lne=`expr $lne + 2`
        tput cup $lne $cl
        echo -n "Content of suspect snapshot are
being recorded, please wait:"
        let lne++
        scrttime=200000
        PROG_BAR
        lne=`expr $lne + 2`
        tput cup $lne $cl

```



```

echo -n "Contents of snapshots gathered."
lne=`expr $lne + 2`
tput cup $lne $cl
echo -n "To proceed for Analysis please
press <ENTER>."

read
tput clear
lne=5
cl=5
tput clear
tput smso
tput cup $lne $cl
echo -n "VM SNAPSHOT ANALYSIS"
lne=`expr $lne + 2`
tput cup $lne $cl
echo -n "View Files with Recent
SETUID/SETGID Changes"

tput rmso
lne=`expr $lne + 2`
tput cup $lne $cl
echo -n "Analysing files, please wait.."
echo "" > $OUTPUT_SETID
echo "" >> $OUTPUT_SETID
echo "      Following are the files with
SETUID/SETGID changes." >> $OUTPUT_SETID
echo "" >> $OUTPUT_SETID
echo "" >> $OUTPUT_SETID
lne=`expr $lne + 2`
tput cup $lne $cl
/bin/sort $DATAFILE_CLEAN > $TEMP_CLEAN &
pid=`echo $!`
sctrtime=50000
PROG_BAR
/bin/sort $DATAFILE_CHANGED > $TEMP_CHANGED
&

pid=`echo $!`
sctrtime=50000
PROG_BAR
/usr/bin/diff $TEMP_CLEAN $TEMP_CHANGED |
grep \> | sed s/\>\ // >> $OUTPUT_SETID &
pid=`echo $!`
sctrtime=50000
PROG_BAR
lne=`expr $lne + 2`
tput cup $lne $cl
echo -n "A list of files with SETUID/SETGID
changes are"

let lne++
tput cup $lne $cl
echo -n "populated and stored at:
$OUTPUT_SETID"

let lne++

```

```

tput cup $lne $cl
echo -n "Please press <ENTER> to view the
file."

read
line_infile=`wc -l $OUTPUT_SETID | awk '{
print $1 }'`

if [ $line_infile -gt 5 ]; then
    tput clear
    tput cup 3 5
    echo -n "Following are the files with
SETUID/SETGID changes."
    tput cup 5 0
    /bin/more $OUTPUT_SETID
    echo ; echo
    GET_CURSOR
else
    lne=`expr $lne + 2`
    tput cup $lne $cl
    tput smso
    echo -n "No files have been
SETUID/SETGID. Lucky us!!"
    tput rmso
fi
lne=`expr $lne + 2`
tput cup $lne $cl
echo -n "Please press <ENTER> to continue."
read
else
    lne=`expr $lne + 2`
    tput cup $lne $cl
    echo "The input is not a number."
    let lne++
    tput cup $lne $cl
    echo -n "Do you wish to continue (y/n): "
    let lne++
    tput cup $lne $cl
    echo -n "Enter N/n to exit."
    line=`expr $lne - 1`
    tput cup $line 36
    read ans
    if [ $ans == n -o $ans == N ] &> /dev/null
; then
        exit
    else
        SECT
    fi
fi
fi
}
SECT
fi

```

**result\_files.sh**

```
#!/bin/bash

function CTRAP {
    trap '
        let lne++
        let lne++
        tput cup $lne $cl
        echo -n "CTRL_C Not Allowed."
        let lne++
        tput cup $lne $cl
        echo -n "Please press <ENTER> to continue."
        ' 2
}

## Get cursor positon
function GET_CURSOR {
    stty -echo
    echo -n $'\e[6n'
    read -d R x; stty echo
    lne=`echo ${x#??} | cut -f1 -d';'`
    lne=`expr $lne - 2`
}

## Get cursor positon

MENUPATH=/home/manish/snapshot_diff
OUTPUTDIR=${MENUPATH}/output_files
TEMP_CLEAN=/tmp/clean
TEMP_CHANGED=/tmp/changed
DATAFILE1=${MENUPATH}/raw_files_path
DATAFILE_CLEAN=${MENUPATH}/files_list_clean
DATAFILE_CHANGED=${MENUPATH}/files_list_changed
OUTPUT_ADDED=${OUTPUTDIR}/files_added.list
OUTPUT_DELETED=${OUTPUTDIR}/files_deleted.list
OUTPUT_EDITED=${OUTPUTDIR}/files_edited.list
OUTPUT_SETID=${OUTPUTDIR}/files_setid.list

choice=1
until [ $choice == 0 ]
do
    lne=5
    cl=5
    tput clear
    tput smso
    tput cup $lne $cl
    echo -n "VM SNAPSHOT ANALYSIS"
    lne=`expr $lne + 2`
    tput cup $lne $cl
    echo -n "View Analysis Result Files"
    tput rmso

```

```

let lne++
i=0
for files in `ls $OUTPUTDIR`
do
    let lne++
    let i++
    no_file[$i]=$files
    tput cup $lne $cl
    echo -n "${i}) $files"
done
lne=`expr $lne + 2`
tput cup $lne $cl
echo -n "0) EXIT"
lne=`expr $lne + 2`
tput cup $lne $cl
echo -n "Please enter choice to view file [1-${i},0]: "
CTRAP; read choice
case $choice in
    [1-${i})
        no_line=`wc -l $OUTPUTDIR/${no_file[$choice]} | awk '{
print $1 }'`
        if [ $no_line -gt 5 ]; then
            tput clear
            more $OUTPUTDIR/${no_file[$choice]}
            GET_CURSOR
        else
            lne=`expr $lne + 2`
            tput cup $lne $cl
            echo -n "File is empty. Either test was not run."
            let lne++
            tput cup $lne $cl
            echo -n "Or the snapshots do not differ."
        fi
        lne=`expr $lne + 2`
        tput cup $lne $cl
        echo -n "Please press <ENTER> to continue."
        read
    ;;
    0) echo ; echo ; exit
    ;;
    *)
        lne=`expr $lne + 2`
        tput cup $lne $cl
        echo -n "Choice Invalid."
        let lne++
        tput cup $lne $cl
        echo -n "Please press <ENTER> to continue."
        read
    ;;
esac
done

```

**check\_md5.sh**

```
#!/bin/bash

function CTRAP {
    trap '
        let lne++
        let lne++
        tput cup $lne $cl
        echo -n "CTRL_C Not Allowed."
        let lne++
        tput cup $lne $cl
        echo -n "Please press <ENTER> to continue."
    ' 2
}

MENUPATH=/home/manish/snapshot_diff
OUTPUTDIR=${MENUPATH}/output_files
TEMP_CLEAN=/tmp/clean
TEMP_CHANGED=/tmp/changed
DATAFILE1=${MENUPATH}/raw_files_path
DATAFILE_CLEAN=${MENUPATH}/files_list_clean
DATAFILE_CHANGED=${MENUPATH}/files_list_changed
OUTPUT_ADDED=${OUTPUTDIR}/files_added.list
OUTPUT_DELETED=${OUTPUTDIR}/files_deleted.list
OUTPUT_EDITED=${OUTPUTDIR}/files_edited.list
OUTPUT_SETID=${OUTPUTDIR}/files_setid.list

choice=1
until [ $choice == 0 ]
do
    lne=5
    cl=5
    tput clear
    tput smso
    tput cup $lne $cl
    echo -n "VM SNAPSHOT ANALYSIS"
    lne=`expr $lne + 2`
    tput cup $lne $cl
    echo -n "Compute MD5 & SHA1 Hashes"
    tput rmso
    lne=`expr $lne + 2`
    tput cup $lne $cl
    echo -n "1) Compute Hash for Files Already Selected"
    let lne++
    tput cup $lne $cl
    echo -n "2) Compute Hash for Another File"
    lne=`expr $lne + 2`
    tput cup $lne $cl
    echo -n "0) EXIT"
    lne=`expr $lne + 2`
    tput cup $lne $cl

```

```

echo -n "Please enter choice [1-2,0]: "
CTRAP; read choice
case $choice in
    1)
        lne=`expr $lne + 2`
        tput cup $lne $cl
        echo -n "Computing MD5 & SHA1 for `sed -n 1p
$DATAFILE1`"
        let lne++
        tput cup $lne $cl
        echo "Please be patient, this may take a while."
        lne=`expr $lne + 2`
        tput cup $lne $cl
        /usr/bin/md5sum `sed -n 1p $DATAFILE1`
        lne=`expr $lne + 2`
        tput cup $lne $cl
        /usr/bin/sha1sum `sed -n 1p $DATAFILE1`
        lne=`expr $lne + 2`
        tput cup $lne $cl
        echo -n "Computing MD5 & SHA1 for `sed -n 2p
$DATAFILE1`"
        let lne++
        tput cup $lne $cl
        echo "Please be patient, this may take a while."
        lne=`expr $lne + 2`
        tput cup $lne $cl
        /usr/bin/md5sum `sed -n 2p $DATAFILE1`
        lne=`expr $lne + 2`
        tput cup $lne $cl
        /usr/bin/sha1sum `sed -n 2p $DATAFILE1`
        lne=`expr $lne + 2`
        tput cup $lne $cl
        echo -n "Please press <ENTER> to continue."
        read

        ;;
    2)

        function PATH_1 {
            lne=`expr $lne + 2`
            tput cup $lne $cl
            echo -n "Please input complete path of file for
MD5 computation"
            let lne++
            tput cup $lne $cl
            echo -n "PATH: "
            read path1
            if [ -z $path1 ]; then
                let lne++
                tput cup $lne $cl
                echo -n "Path cannot be left empty."
                let lne++

```

```

tput cup $lne $cl
echo -n "Do you wish to continue (y/n): "
let lne++
tput cup $lne $cl
echo -n "Enter N/n to exit."
line=`expr $lne - 1`
tput cup $line 36
read ans
if [ $ans == n -o $ans == N ] &> /dev/null
; then
    exit
else
    PATH_1
fi
else
if [ -f $path1 ]; then
    lne=`expr $lne + 2`
    tput cup $lne $cl
    echo -n "Computing MD5 & SHA1 for
$path1"

    let lne++
    tput cup $lne $cl
    echo "Please be patient, this may take
a while."

    lne=`expr $lne + 2`
    tput cup $lne $cl
    /usr/bin/md5sum $path1
    lne=`expr $lne + 2`
    tput cup $lne $cl
    /usr/bin/sha1sum $path1
else
    let lne++
    tput cup $lne $cl
    echo -n "File does not exist or path
incorrect."

    let lne++
    tput cup $lne $cl
    echo -n "Do you wish to continue
(y/n): "

    let lne++
    tput cup $lne $cl
    echo -n "Enter N/n to exit."
    line=`expr $lne - 1`
    tput cup $line 36
    read ans
    if [ $ans == n -o $ans == N ] &>
/dev/null ; then
        exit
    else
        PATH_1
    fi
fi

```

```
                fi
            }
            PATH_1
            lne=`expr $lne + 2`
            tput cup $lne $cl
            echo -n "Please press <ENTER> to continue."
            read
        ;;
    0) echo ; echo ; exit
        ;;
    *)
        lne=`expr $lne + 2`
        tput cup $lne $cl
        echo -n "Choice Invalid."
        let lne++
        tput cup $lne $cl
        echo -n "Please press <ENTER> to continue."
        read
    ;;
esac
done
```