# Appendix B: Procedures to Conduct the Tools Evaluation and Collect Data

## **B.1 Test case 01: FIT-ID-01**

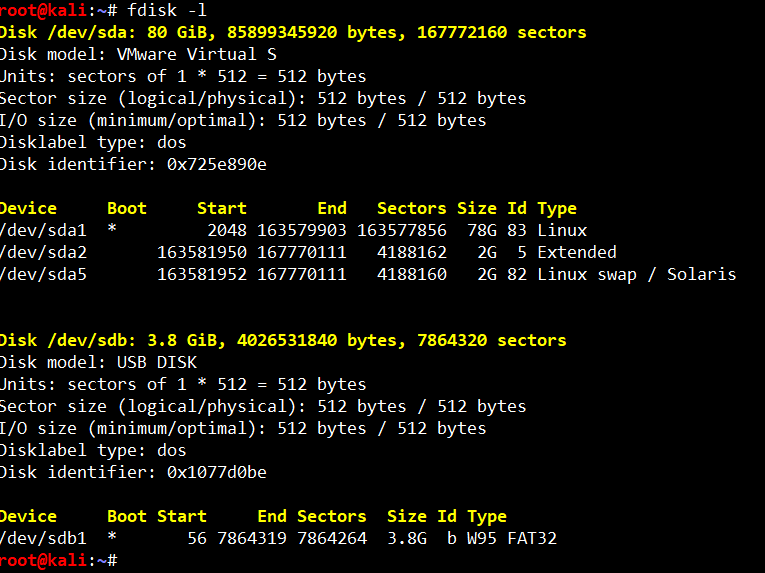
### **B.1.1 Dc3dd**

**Step 1**

List all the hard disks or USB devices

**/dev/sdc** is the USB device

|  |
| --- |
| fdisk -l |



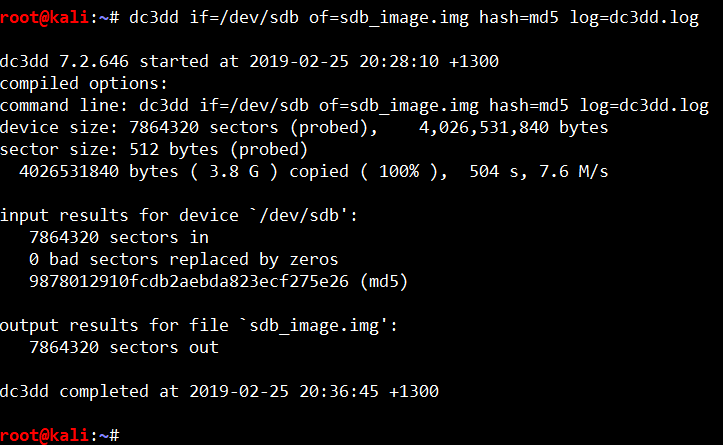
**Step 2**

Create the forensic image

|  |
| --- |
| cd /media/investigator/imagedrive |

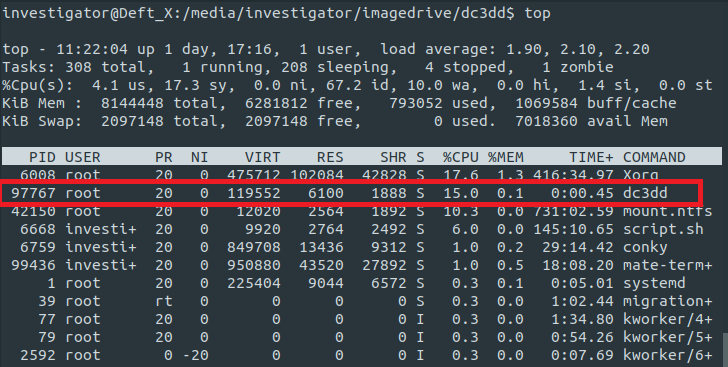
|  |
| --- |
| sudo dc3dd if=/dev/sdc hof=image.img hash=sha256 log=dc3dd.log hlog=imageName-hash.log verb=on |

* **if**  - input file
* **/dev/sdc** - source /suspect drive (whole disk)
* **of -** output file
* **image.raw** - name of the image file
* **hash** - Definition of hash algorithms
* **log** - Path of the log file



**Step 3**

At the same time run the top command to find out the task ID



Then run the command to log the hardware resources usage by the tool

|  |
| --- |
| pidstat -h -r -u -v -p 97767 30 >> hardware-usage.txt |

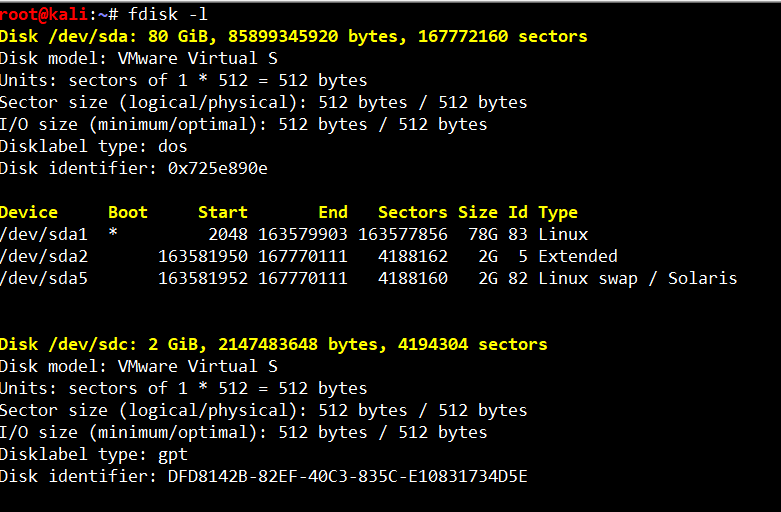
This command is used to monitor this process IDs 97767 every 30 seconds use and write all the log into hardware-log.txt

### **B.1.2 Dcfldd**

**Step 1**

Find the disk on the machine

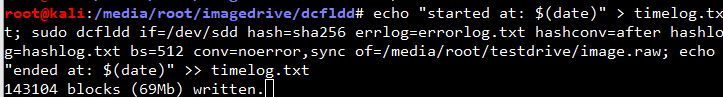
|  |
| --- |
| fdisk -l |



**Step 2**

Select the disk want to create the image then run the command

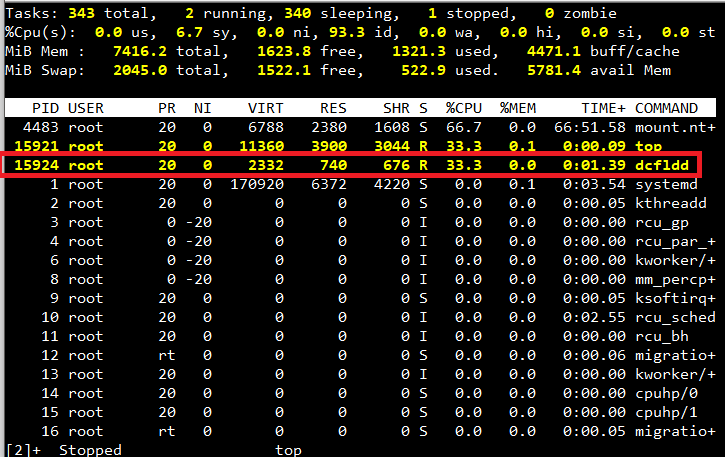
|  |
| --- |
| echo "started at: $(date)" > timelog.txt; sudo dcfldd if=/dev/sdc hash=sha256 errlog=errorlog.txt hashconv=after hashlog=hashlog.txt bs=512 conv=noerror,sync of=image.raw; echo "ended at: $(date)" >> timelog.txt |



* **echo "started at: $(date)" and echo "ended at: $(date)" >> timelog.txt -** Log the time start and finish the command
* **if**  - input file
* **/dev/sdc** - source /suspect drive (whole disk)
* **of -** output file
* **image.raw** - name of the image file
* **hash** - Definition of hash algorithms
* **log** - Path of the log file

**Step 3**

At the same time run the top command to find out the task ID



Then run the command to log the hardware resources usage by the tool

|  |
| --- |
| pidstat -h -r -u -v -p 15924 30 >> hardware-usage.txt |

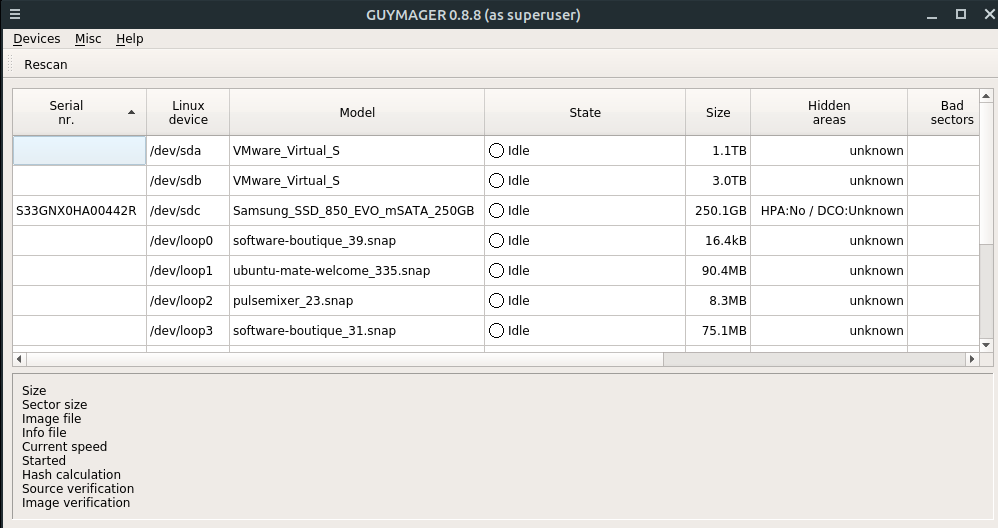


### **B.1.3 Guymager**

**Step 1**

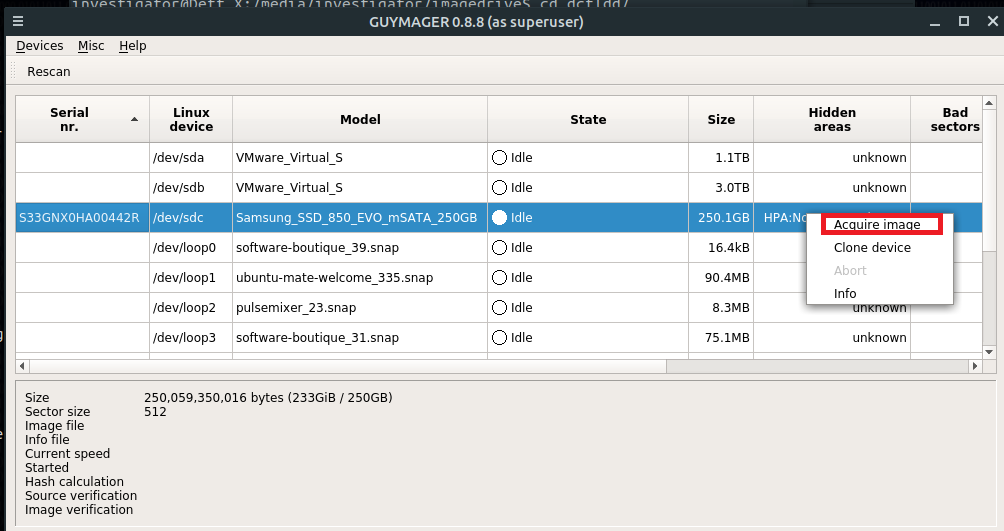
Start the tool by using the command below

|  |
| --- |
| sudo guymager |



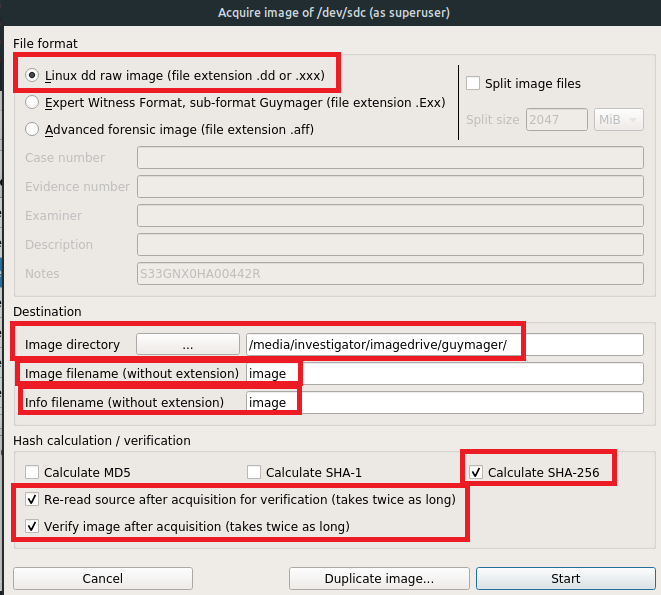
**Step 2**

Select the hard drive to create the forensic image. Then right click and select **Acquire Image**

****

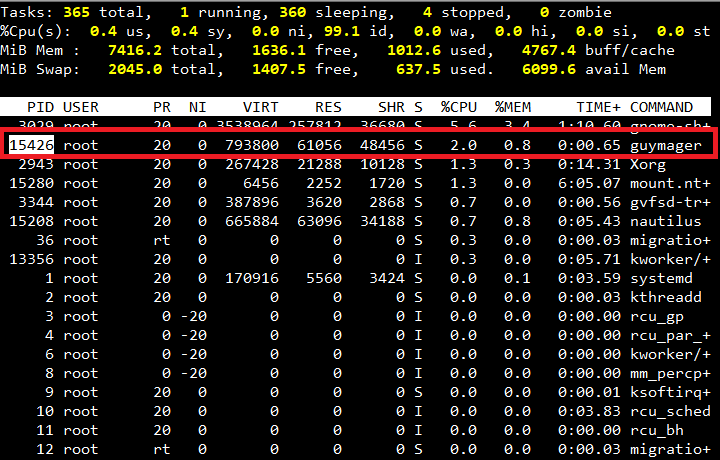
**Step 3**

Select the forensic file format then select the size of the split forensic images. Select the output destination and then select the hash values verification



**Step 4**

At the same time run the top command to find out the task ID



Then run the command to log the hardware resources usage by the tool

|  |
| --- |
| pidstat -h -r -u -v -p 15426 30 >> hardware-usage.txt |

This command is used to monitor this process IDs 15426 every 30 seconds use and write all the log into hardware-log.txt



## **B.2 Test case 02: FIT-ID-02**

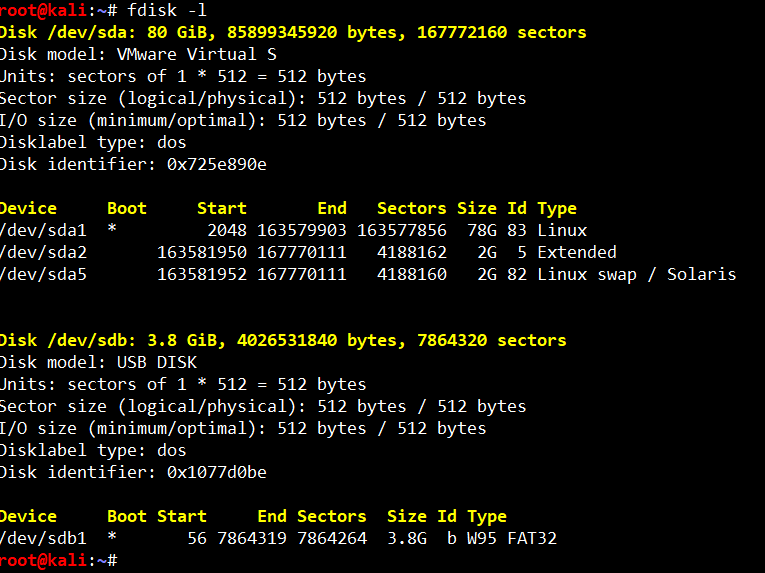
### **B.2.1 Dc3dd**

**Step 1**

List all the hard disk or USB devices

**/dev/sdc** is the USB device

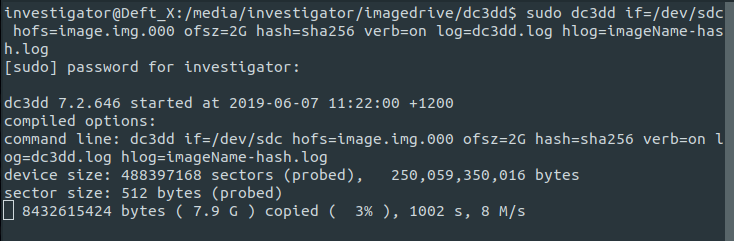
|  |
| --- |
| fdisk -l |



**Step 2**

Get an image of the drive. 2GB (2 \* 1024 \*1024 \*1024 bytes) chunks. The hash value for each and for a total. Log all hashes and activity

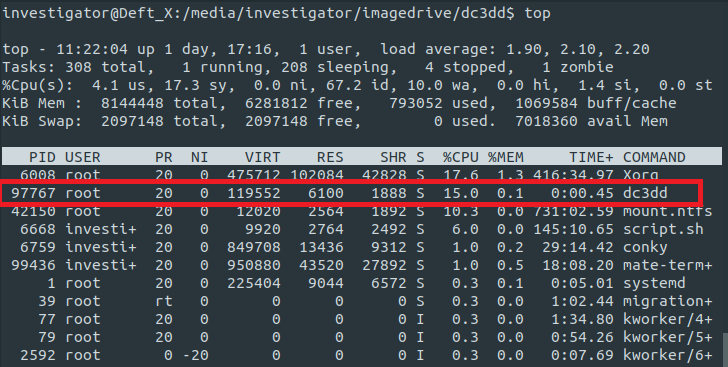
|  |
| --- |
| sudo dc3dd if=/dev/sdc hofs=image.img.000 ofsz=2G hash=sha256 verb=on log=dc3dd.log hlog=imageName-hash.log |



* **if** – input file or device
* **hofs** – substitute for of or ofs – h(ash)o(output)f(ile)s(split) -.000 is the extension pattern of the output chunks
* **ofsz** – output file size or size of chunks. 2G means 2\*1024\*2024\*1024 (MB or GB would be 1000 vs 1024 multipliers)
* **hash** – set hash algorithms. Can use one or more with hofs option
* **verb** – verbose output
* **log** – operational log – note bad sectors and what-not
* **hlog** – hash log – logs total and piece-wise hashes

**Step 3**

At the same time run the top command to find out the task ID



Then run the command to log the hardware resources usage by the tool

|  |
| --- |
| pidstat -h -r -u -v -p 97767 30 >> hardware-usage.txt |

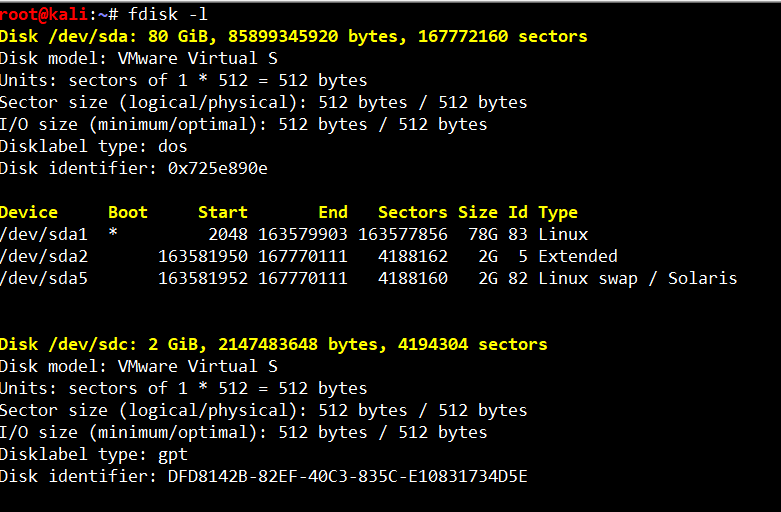
This command is used to monitor this process IDs 97767 every 30 seconds use and write all the log into hardware-log.txt

### **B.2.2 Dcfldd**

**Step 1**

Find the disk on the machine

|  |
| --- |
| fdisk -l |

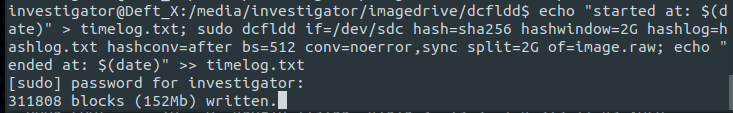


**Step 2**

Select the disk want to create the image then run the command

|  |
| --- |
| echo "started at: $(date)" > timelog.txt; sudo dcfldd if=/dev/sdc hash=sha256 hashwindow=2G hashlog=hashlog.txt hashconv=after bs=512 conv=noerror,sync split=2G of=image.raw; echo "ended at: $(date)" >> timelog.txt |

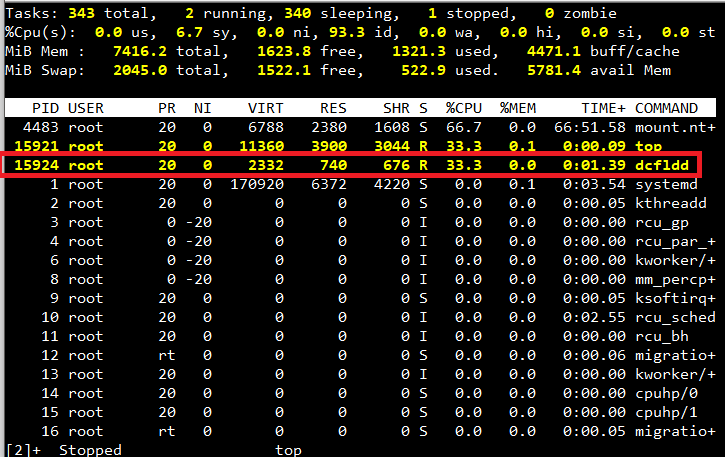
Get an image of the drive. 2GB (2 \* 1024 \*1024 \*1024 bytes) chunks. The hash value for each and total. Log all hashes and activity



* **echo "started at: $(date)" and echo "ended at: $(date)" >> timelog.txt -** Log the time start and finish the command
* **if**  - input file
* **/dev/sdc** - source /suspect drive (whole disk)
* **of -** output file
* **image.raw** - the name of the image file
* **hash** - Definition of hash algorithms
* **log** - Path of the log file

**Step 3**

At the same time run the top command to find out the task ID



Then run the command to log the hardware resources usage by the tool

|  |
| --- |
| pidstat -h -r -u -v -p 15924 30 >> hardware-usage.txt |

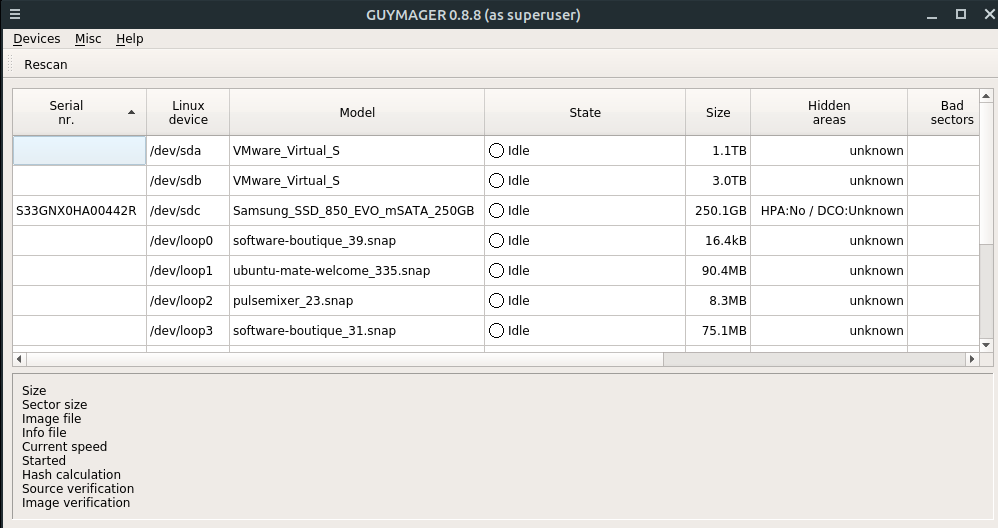


### **B.2.3 Guymager**

**Step 1**

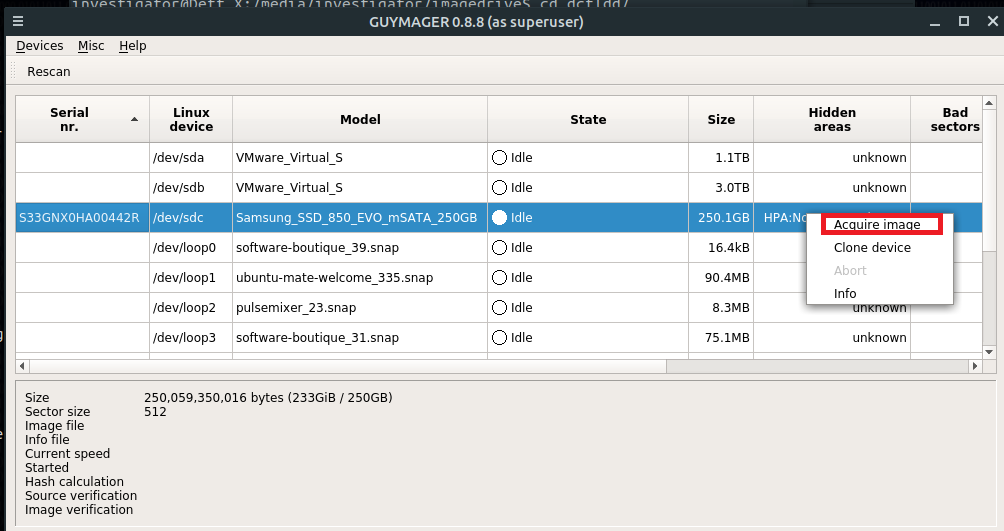
Start the tool by using the command below

|  |
| --- |
| sudo guymager |



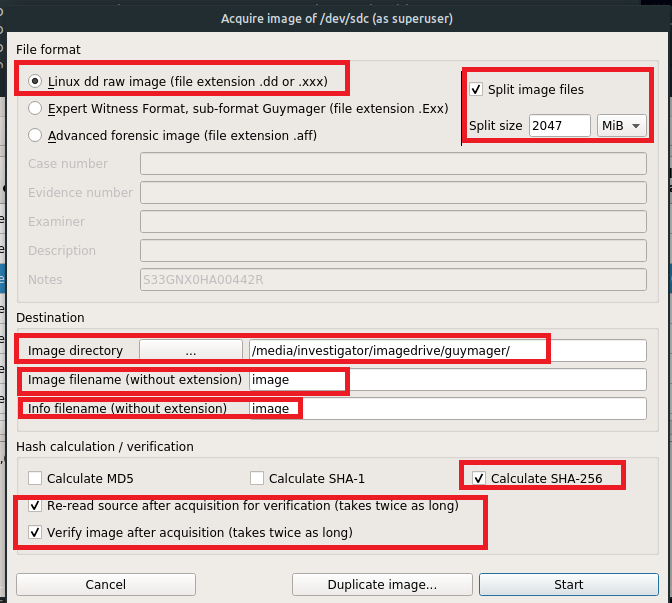
**Step 2**

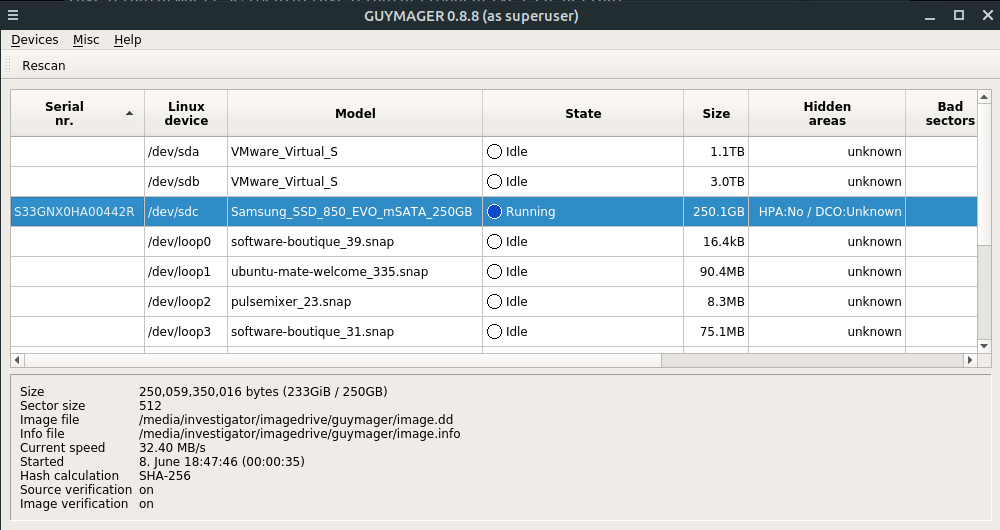
Select the hard drive to create a forensic image. Then right click and select **Acquire Image**

****

**Step 3**

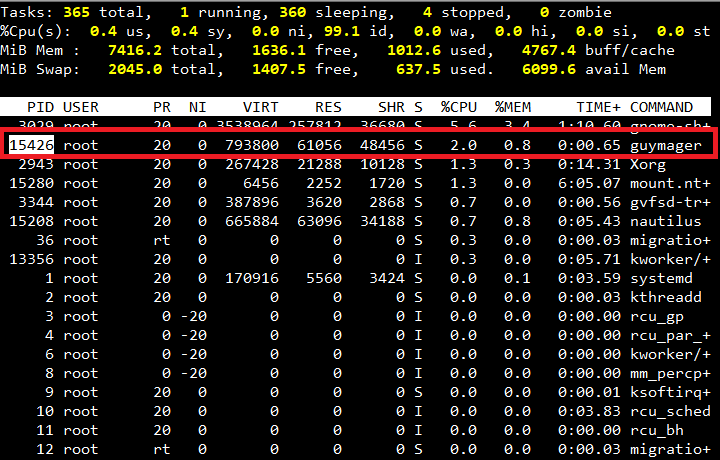
Select the forensic file format then select the size of the split forensic images. Select the output destination and then select the hash values verification





**Step 4**

At the same time run the top command to find out the task ID



Then run the command to log the hardware resources usage by the tool

|  |
| --- |
| pidstat -h -r -u -v -p 15426 30 >> hardware-usage.txt |

This command is used to monitor this process IDs 15426 every 30 seconds use and write all the log into hardware-log.txt



## **B.3 Test case 03: FIT-ID-03**

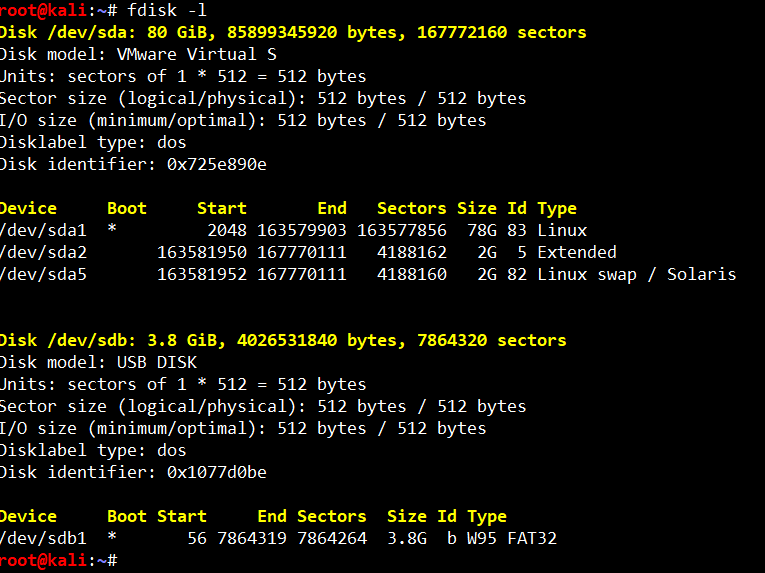
### **B.3.1 Dc3dd**

**Step 1**

List all the hard disk or USB devices

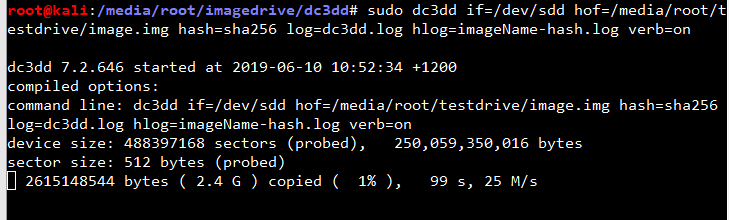
**/dev/sdc** is the USB device

|  |
| --- |
| fdisk -l |



**Step 2**

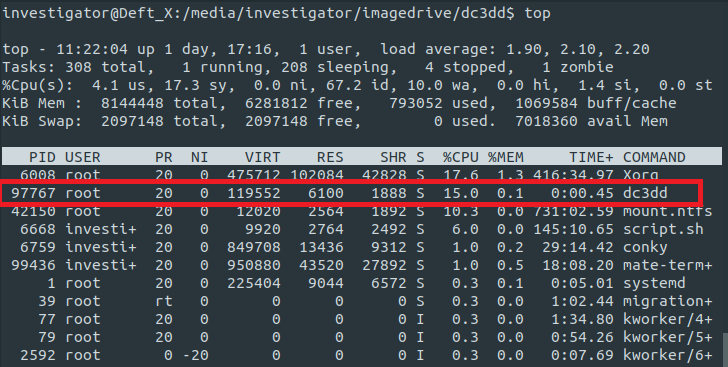
|  |
| --- |
| sudo dc3dd if=/dev/sdd hof=/media/root/testdrive/image.img hash=sha256 log=dc3dd.log hlog=imageName-hash.log verb=on |



* **if** – input file or device
* **hash** – set hash algorithms. Can use one or more with hofs option
* **verb** – verbose output
* **log** – operational log – note bad sectors and what-not
* **hlog** – hash log – logs total and piece-wise hashes

**Step 3**

At the same time run the top command to find out the task ID



Then run the command to log the hardware resources usage by the tool

|  |
| --- |
| pidstat -h -r -u -v -p 97767 30 >> hardware-usage.txt |

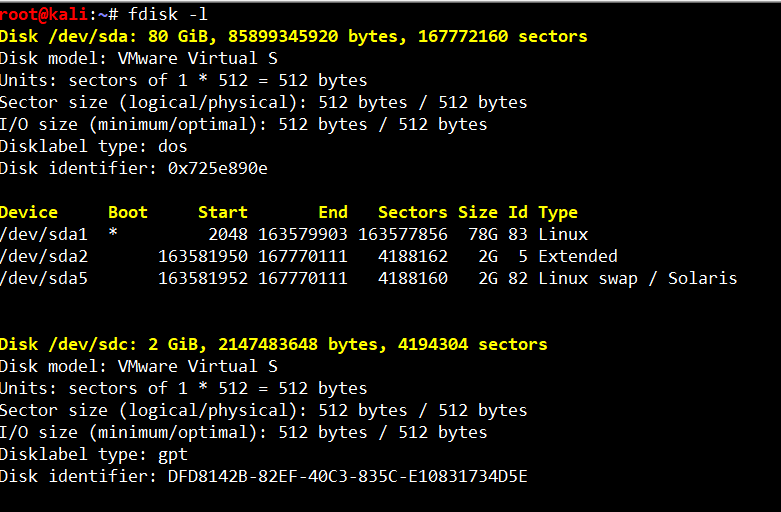
This command is used to monitor this process IDs 97767 every 30 seconds use and write all the log into hardware-log.txt.

### **B.3.2 Dcfldd**

**Step 1**

Find the disk on the machine

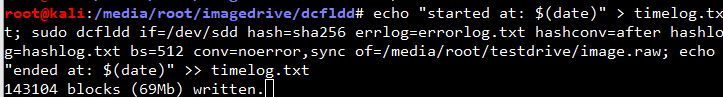
|  |
| --- |
| fdisk -l |



**Step 2**

Select the disk want to create the image then run the command

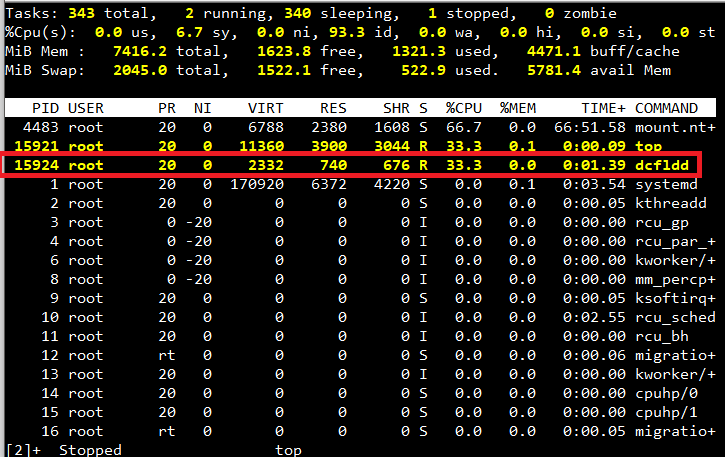
|  |
| --- |
| echo "started at: $(date)" > timelog.txt; sudo dcfldd if=/dev/sdc hash=sha256 errlog=errorlog.txt hashconv=after hashlog=hashlog.txt bs=512 conv=noerror,sync of=image.raw; echo "ended at: $(date)" >> timelog.txt |



* **echo "started at: $(date)" and echo "ended at: $(date)" >> timelog.txt -** Log the time start and finish the command
* **if**  - input file
* **/dev/sdc** - source /suspect drive (whole disk)
* **of -** output file
* **image.raw** - name of the image file
* **hash** - Definition of hash algorithms
* **log** - Path of the log file

**Step 3**

At the same time run the top command to find out the task ID



Then run the command to log the hardware resources usage by the tool

|  |
| --- |
| pidstat -h -r -u -v -p 15924 30 >> hardware-usage.txt |

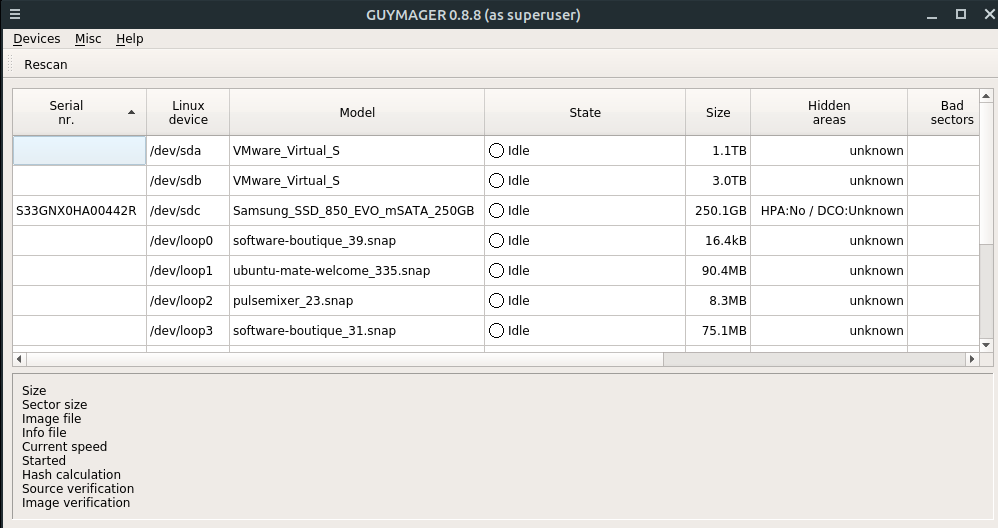


### **B.3.3 Guymager**

**Step 1**

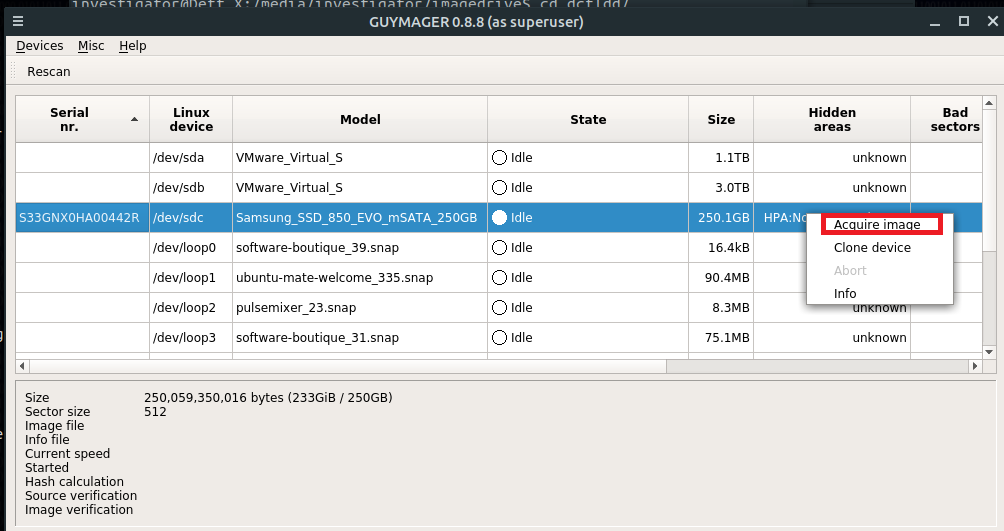
Start the tool by using the command below

|  |
| --- |
| sudo guymager |



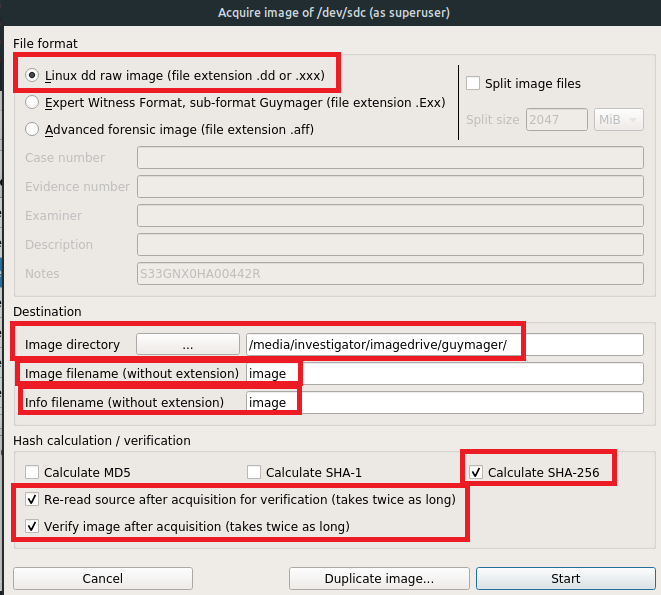
**Step 2**

Select the hard drive to create a forensic image. Then right click and select **Acquire Image**

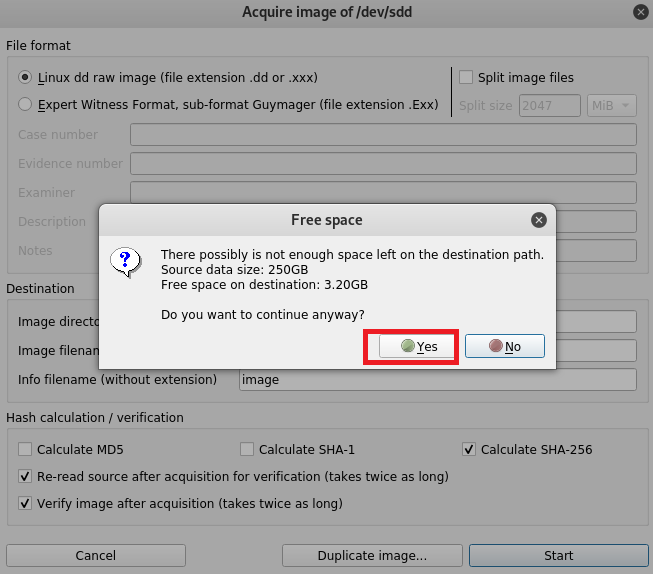
****

**Step 3**

Select the forensic file format then select the size of the split forensic images. Select the output destination and then select the hash values verification

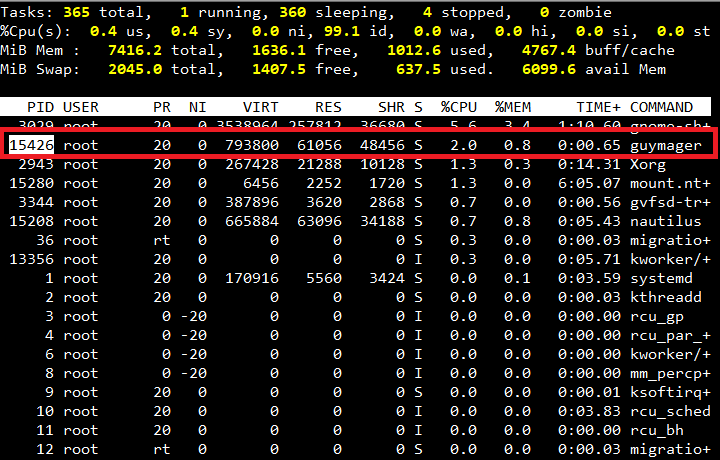


A window will pop up notify the output location will not have enough storage space. Click Yes



**Step 4**

At the same time run the top command to find out the task ID



Then run the command to log the hardware resources usage by the tool

|  |
| --- |
| pidstat -h -r -u -v -p 15426 30 >> hardware-usage.txt |

This command is used for to monitor this process IDs 15426 every 30 seconds use and write all the log into hardware-log.txt



## **B.4 Test case 04: FIT-ID-04**

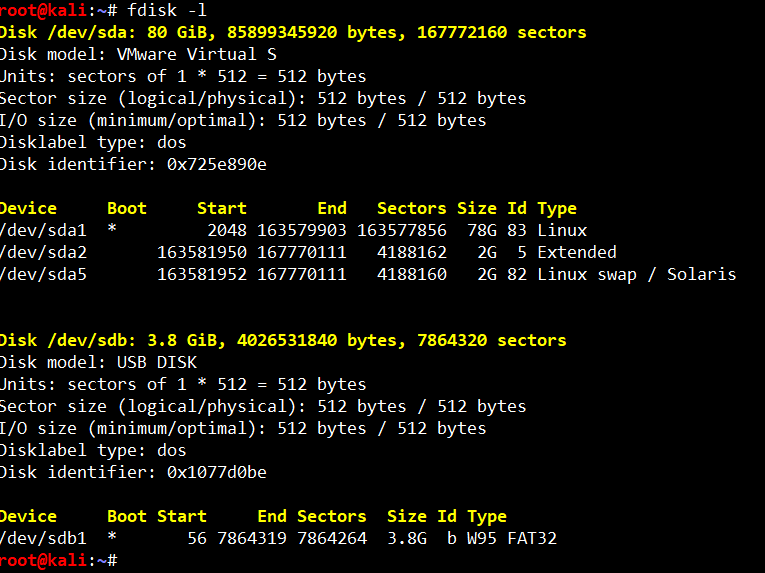
### **B.4.1 Dc3dd**

**Step 1**

List all the hard disk or USB devices

**/dev/sdc** is the USB device

|  |
| --- |
| fdisk -l |



**Step 2**

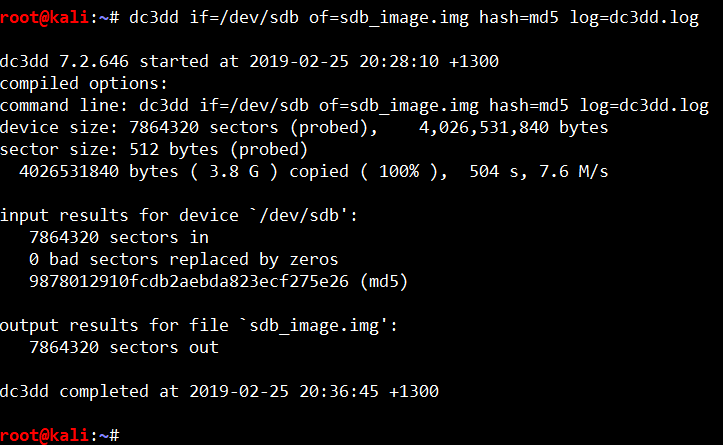
Create the image

For more details: <https://tools.kali.org/forensics/dc3dd>

|  |
| --- |
| cd /media/investigator/imagedrive |

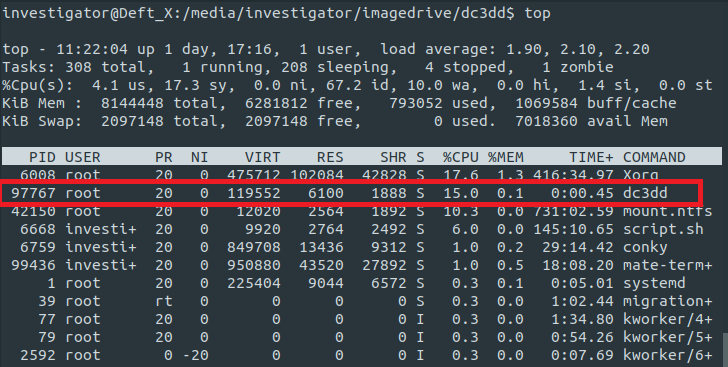
|  |
| --- |
| sudo dc3dd if=/dev/sdc hof=image.img hash=sha256 log=dc3dd.log hlog=imageName-hash.log verb=on |

* **if**  - input file
* **/dev/sdc** - source /suspect drive (whole disk)
* **of -** output file
* **image.raw** - name of the image file
* **hash** - Definition of hash algorithms
* **log** - Path of the log file



**Step 3**

At the same time run the top command to find out the task ID



Then run the command to log the hardware resources usage by the tool

|  |
| --- |
| pidstat -h -r -u -v -p 97767 30 >> hardware-usage.txt |

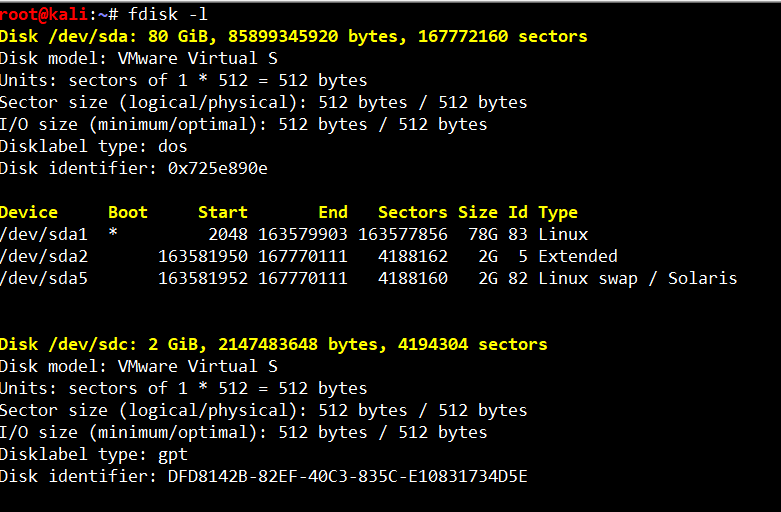
This command is used to monitor this process IDs 97767 every 30 seconds use and write all the log into hardware-log.txt

### **B.4.2 Dcfldd**

**Step 1**

Find the disk on the machine

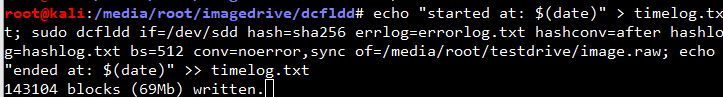
|  |
| --- |
| fdisk -l |



**Step 2**

Select the disk want to create the image then run the command

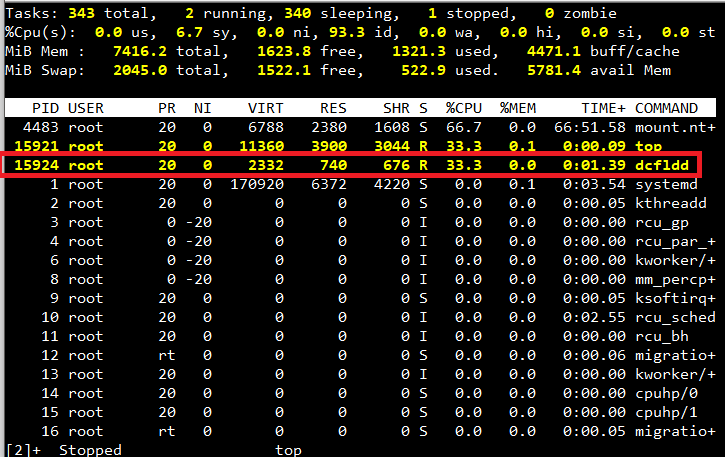
|  |
| --- |
| echo "started at: $(date)" > timelog.txt; sudo dcfldd if=/dev/sdc hash=sha256 errlog=errorlog.txt hashconv=after hashlog=hashlog.txt bs=512 conv=noerror,sync of=image.raw; echo "ended at: $(date)" >> timelog.txt |



* **echo "started at: $(date)" and echo "ended at: $(date)" >> timelog.txt -** Log the time start and finish the command
* **if**  - input file
* **/dev/sdc** - source /suspect drive (whole disk)
* **of -** output file
* **image.raw** - name of the image file
* **hash** - Definition of hash algorithms
* **log** - Path of the log file

**Step 3**

At the same time run the top command to find out the task ID



Then run the command to log the hardware resources usage by the tool

|  |
| --- |
| pidstat -h -r -u -v -p 15924 30 >> hardware-usage.txt |

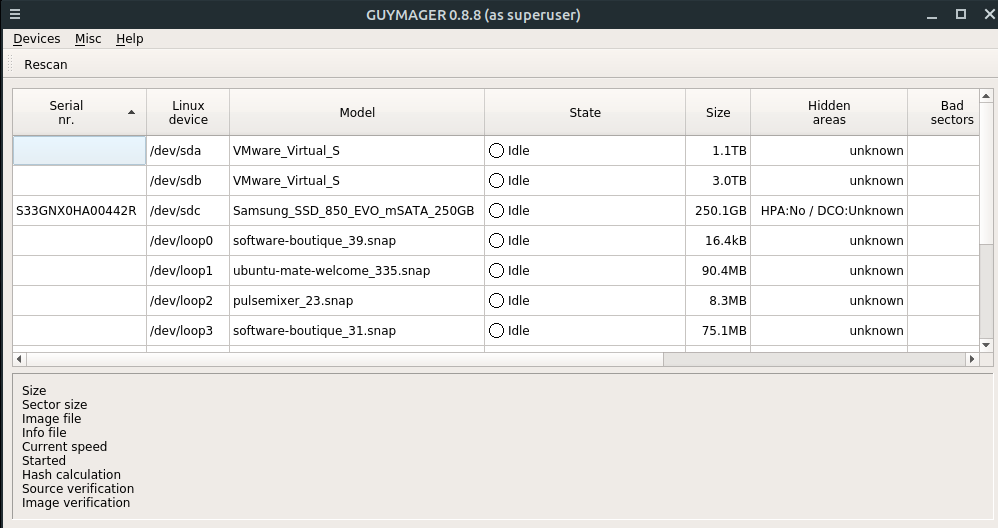


### **B.4.3 Guymager**

**Step 1**

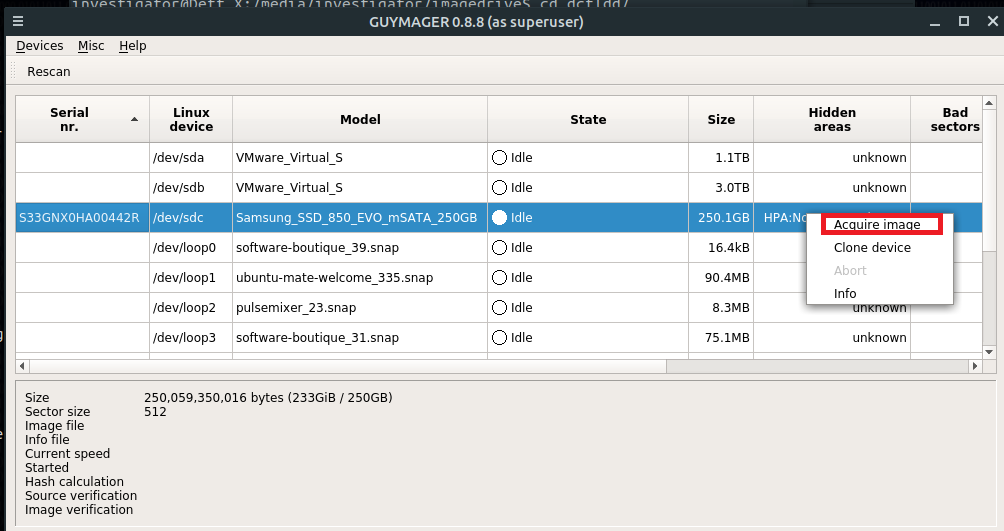
Start the tool by using the command below

|  |
| --- |
| sudo guymager |



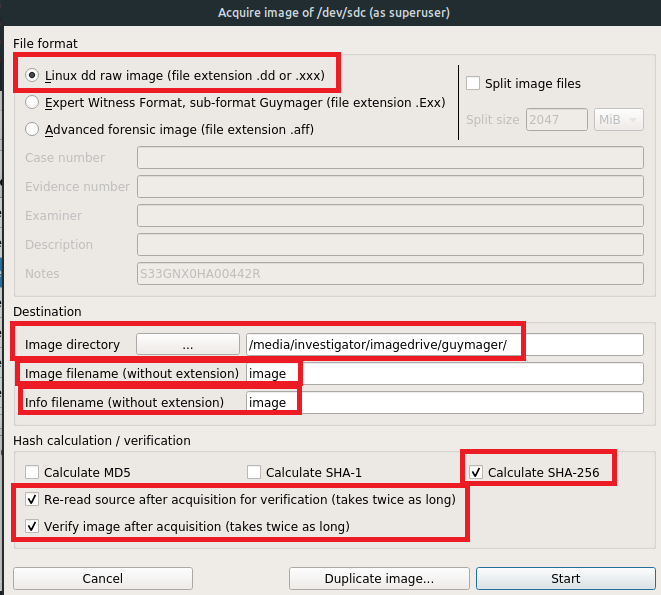
**Step 2**

Select the hard drive to create a forensic image. Then right click and select **Acquire Image**

****

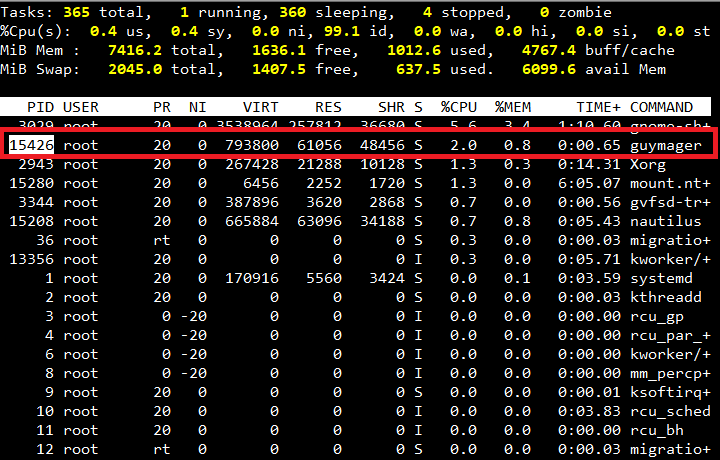
**Step 3**

Select the forensic file format then select the size of the split forensic images. Select the output destination and then select the hash values verification



**Step 4**

At the same time run the top command to find out the task ID



Then run the command to log the hardware resources usage by the tool

|  |
| --- |
| pidstat -h -r -u -v -p 15426 30 >> hardware-usage.txt |

This command is used to monitor this process IDs 15426 every 30 seconds use and write all the log into hardware-log.txt

