Forensic Data Carving using Foremost

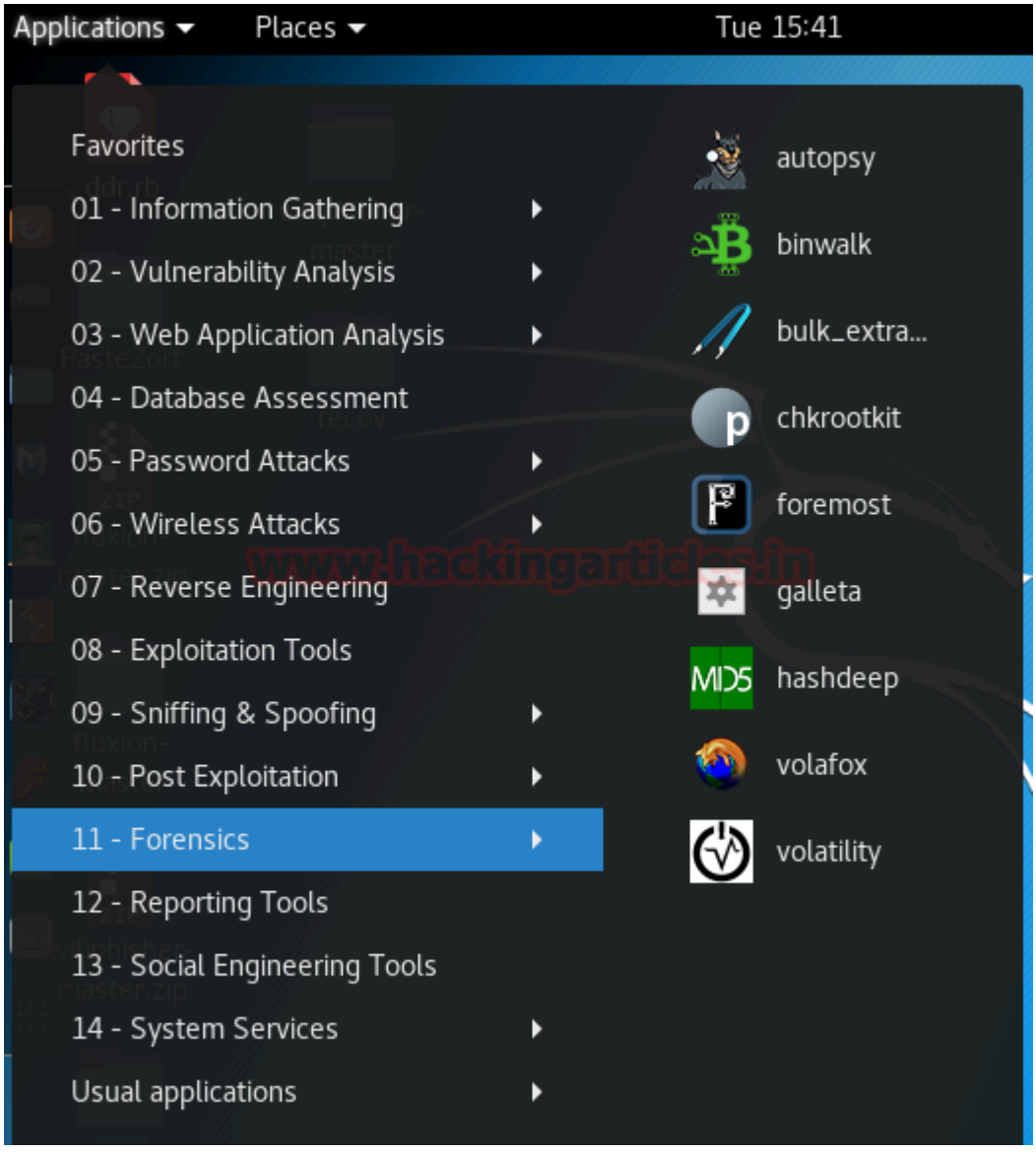
January 13, 2018 By Raj Chandel

Foremost is a program that is used to carve data from disk image files, it is an extremely useful tool and very easy to use.

For the purpose of this article we have used an Ubuntu disk image file and the process has been repeated twice. The purpose of doing so was to see if Foremost can carve data out of incomplete disk images as well. We have used Kali Linux but if you want you can install Foremost on pretty much any distro of Linux.

Here's how it was done:

Navigate to the A**pplications** menu in Kali, **Forensics** is option 11. The fifth option from top in the **Forensics** menu is **Foremost.** Click on it and let's get to carving some data!!



Foremost starts and shows you the options you have at your disposal.

```
foremost version 1.5.7 by Jesse Kornblum, Kris Kendall, and Nick Mikus.
$ foremost [-v|-V|-h|-T|-Q|-q|-a|-w-d] [-t <type>] [-s <blocks>] [-k <size>]
        [-b <size>] [-c <file>] [-o <dir>] [-i <file]

-V  - display copyright information and exit
-t  - specify file type.  (-t jpeg,pdf ...)
-d  - turn on indirect block detection (for UNIX file-systems)
-i  - specify input file (default is stdin)
-a  - Write all headers, perform no error detection (corrupted files)
-w  - Only write the audit file, do not write any detected files to the disk
-o  - set output directory (defaults to output)
-c  - set configuration file to use (defaults to foremost.conf)
-q  - enables quick mode. Search are performed on 512 byte boundaries.
-Q  - enables quiet mode. Suppress output messages.
-v  - verbose mode. Logs all messages to screen
```

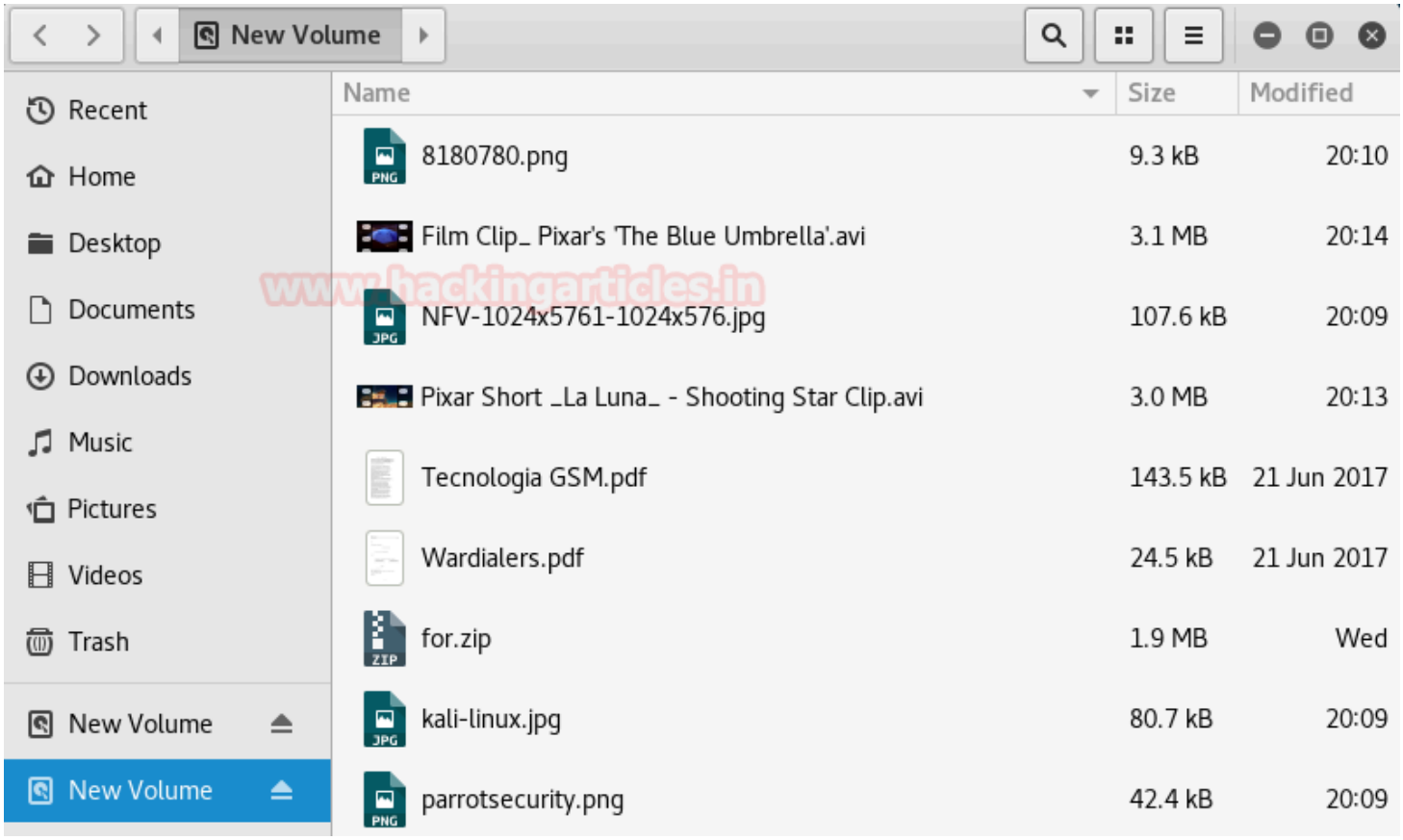
In order to keep things simple, you first want to navigate to the **Desktop** using **"cd Desktop"**.

Next, make a folder on the desktop by the name of **"recov".** This isn't a mandatory step, it just makes things easier to access by making a new folder where the carved data will be stored.

```
root@kali:~# cd Desktop
root@kali:~/Desktop# mkdir recov
```

We will be dealing with the disk image of a flash drive partition, so let's make one using the **"dd"** command**.** The **dd** command can be used to copy files and with the option of converting the data format in the process.

In the interest of thoroughness we have copied .docx, .jpeg, .png, .zip, .pdf and .avi files onto the partition from which we will be making our disk image.



Now let's make a disk image.

In a new terminal window, type the following **"fdisk –l | grep /dev/"**. This command will show you the disk partitions available to you without any clutter.

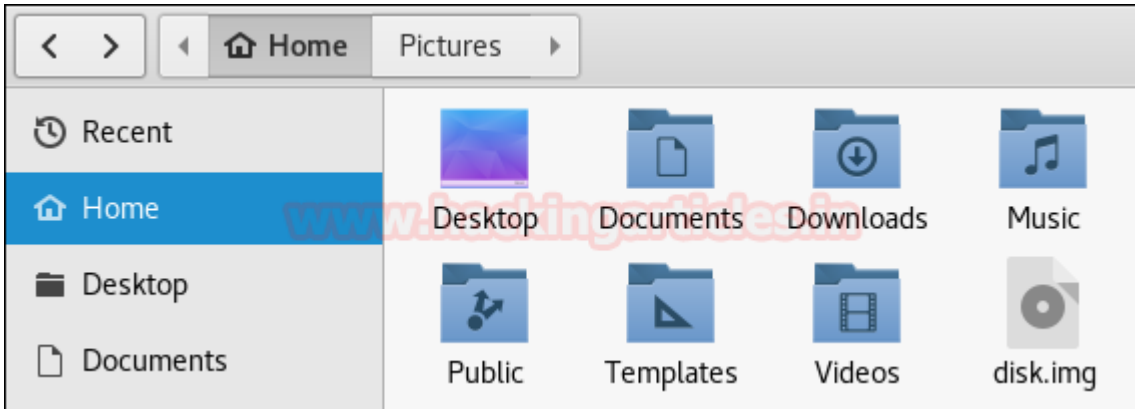
```
root@kali:~# fdisk -l | grep /dev/
Disk /dev/sda: 20 GiB, 21474836480 bytes, 41943040 sectors
/dev/sda1  *        2048 37750783 37748736  18G 83 Linux
/dev/sda2       37752830 41940991  4188162   2G  5 Extended
/dev/sda5       37752832 41940991  4188160   2G 82 Linux swap / Solaris
Disk /dev/sdb: 15 GiB, 16047407104 bytes, 31342592 sectors
/dev/sdb1  *          63 27148287 27148225   13G  7 HPFS/NTFS/exFAT
/dev/sdb2       27148288 27353087   204800  100M  7 HPFS/NTFS/exFAT
/dev/sdb3       27353088 31340543  3987456  1.9G  7 HPFS/NTFS/exFAT
```

The partition we are concerned with is **/dev/sbd2**, this was specially allocated **10 MB** of space so that the imaging process is quick**.**

The command to create the disk image is **"dd if=/dev/sdb2 of=disk.img"**. Here, **"dd"** is the utility we are using, **"if="** is to denote the input destination and **"of="** is to denote the output destination and name of the image file we are creating.

```
root@kali:~# dd if=/dev/sdb2 of=disk.img
204800+0 records in
204800+0 records out
104857600 bytes (105 MB, 100 MiB) copied, 7.70721 s, 13.6 MB/s
```

We have not specified any output destination, but, just the name for the image file. The image file will be created in the **Home** directory by default. Copy the disk image file from here and place it on the desktop.



Let's navigate back to the terminal where we have **Foremost** running and start the file carving process.

This disk image file will be carved for .**jpeg, .png, .zip, .pdf and .avi** file formats. We will not be instructing **Foremost** to carve the **.docx** but, since one exists in the **.zip** we have placed inside the disk image, it will do so automatically.

Type the following **"foremost -t jpeg,png,zip,pdf,avi -i disk.img -o recov –v"**.

To break this down **"-t"** is setting the file types we want to carve out of the disk image, here those are .jpeg and .png.

"-i" is specifying the input file, the **"disk.img"** that is placed on the desktop.

"-o" is telling Foremost where we want the carved files to be stored, for that we have the **"recov"** folder on the desktop that we made earlier.

"-v" is to tell Foremost to log all the messages that appear on screen as the file is being carved into a text file in the output folder (recov) as an audit report.

```
root@kali:~/Desktop# foremost -t jpeg,png,zip,pdf,avi -i disk.img -o recov -v
```

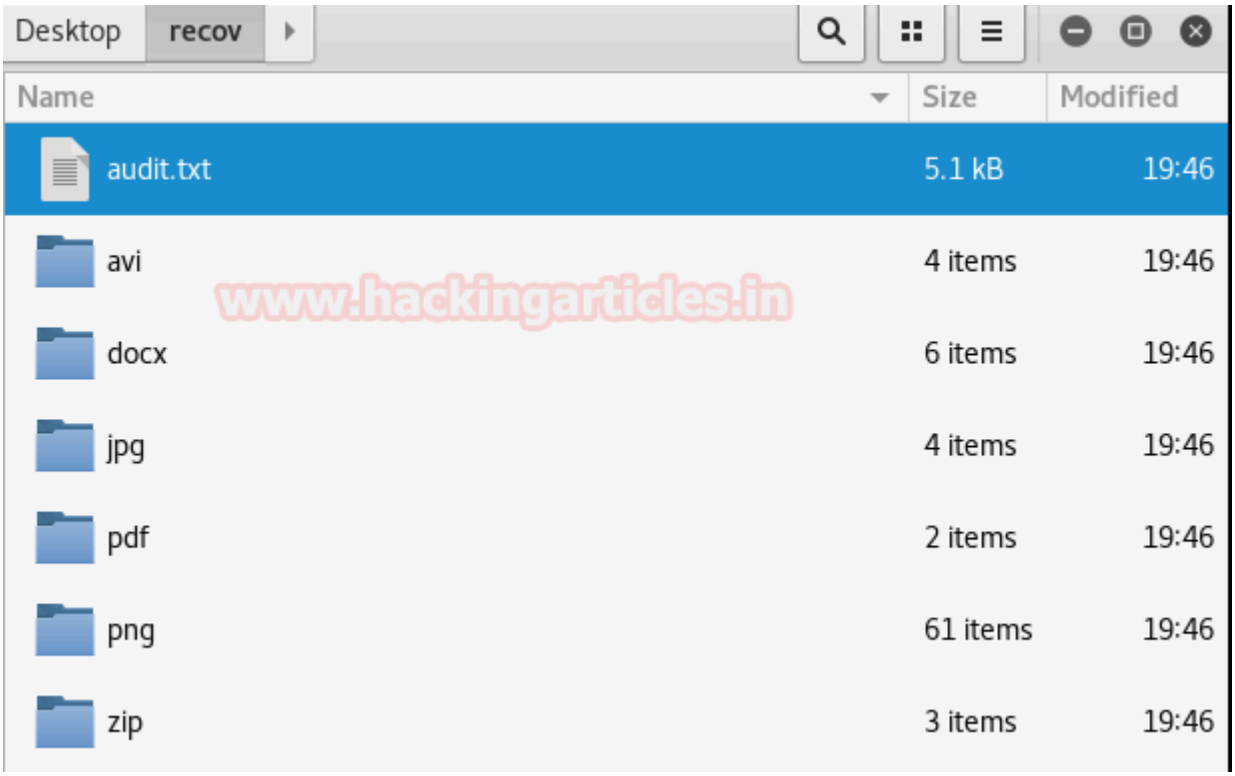
That's all it takes for Foremost to start digging into the disk image. The process looks like this.

```
Foremost version 1.5.7 by Jesse Kornblum, Kris Kendall, and Nick Mikus
Audit File

Foremost started at Fri Jan 12 19:45:59 2018
Invocation: foremost -t jpeg,png,zip,pdf,avi -i disk.img -o recov -v
Output directory: /root/Desktop/recov
Configuration file: /etc/foremost.conf
Processing: disk.img
|------------------------------------------------------------------
File: disk.img
Start: Fri Jan 12 19:45:59 2018
Length: 100 MB (104857600 bytes)

Num      Name (bs=512)         Size      File Offset     Comment

0:      00021768.jpg          78 KB        11145216
1:      00021928.jpg         105 KB        11227136
2:      00192128.jpg         105 KB        98369536
3:      00198208.jpg          78 KB       101482496
4:      00011328.png           9 KB         5799936       (400 x 400)
5:      00011372.png          32 KB         5822809       (613 x 415)
6:      00011436.png           9 KB         5855655       (792 x 251)
```

Once Foremost is done carving the disk image, it shows you the result: that's is, how many of which file types have been carved. All it took was a second, to get the job done.

```
76:     00011904.avi           2 MB         6094848
77:     00022928.avi           2 MB        11739136
78:     00127136.avi           2 MB        65093632
79:     00159360.avi           2 MB        81592320
*|
Finish: Fri Jan 12 19:46:00 2018

80 FILES EXTRACTED

jpg:= 4
png:= 61
zip:= 9
pdf:= 2
avi:= 4
------------------------------------------------------------------

Foremost finished at Fri Jan 12 19:46:00 2018
```

Now open the output (recov) folder and you will see an audit report and six folders which will be named by the file types we invoked Foremost to carve for us.



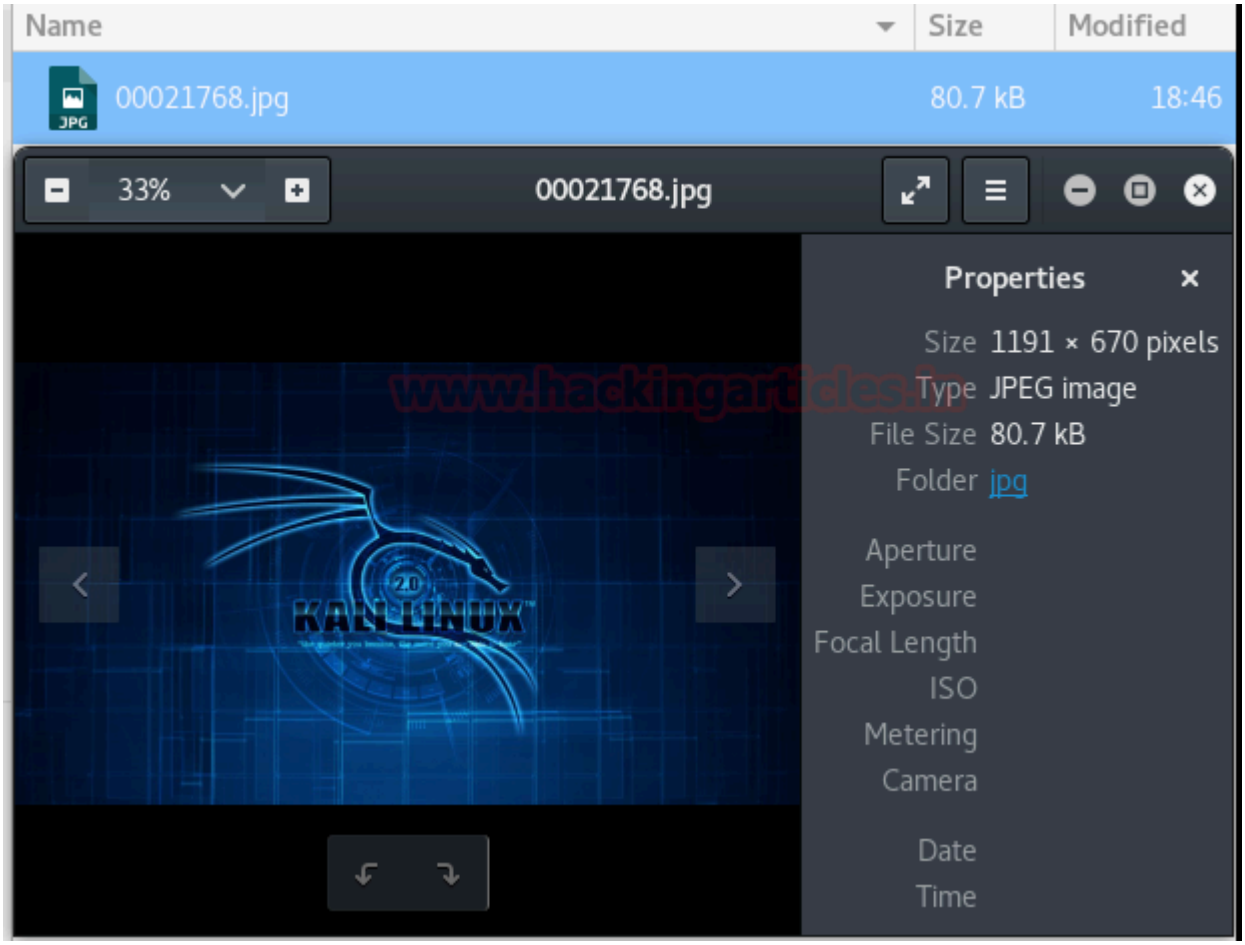
First, the audit report. It shows us the particulars of the scan, which file types were carved, from which image file, the size of the image file, where it was located, where the output folder was located, etc. Let's have a look.

```
Foremost version 1.5.7 by Jesse Kornblum, Kris Kendall, and Nick Mikus
Audit File

Foremost started at Fri Jan 12 19:45:59 2018
Invocation: foremost -t jpeg,png,zip,pdf,avi -i disk.img -o recov -v
Output directory: /root/Desktop/recov
Configuration file: /etc/foremost.conf
------------------------------------------------------------------
File: disk.img
Start: Fri Jan 12 19:45:59 2018
Length: 100 MB (104857600 bytes)

Num      Name (bs=512)         Size      File Offset     Comment

0:      00021768.jpg          78 KB        11145216
1:      00021928.jpg         105 KB        11227136
2:      00192128.jpg         105 KB        98369536
3:      00198208.jpg          78 KB       101482496
```

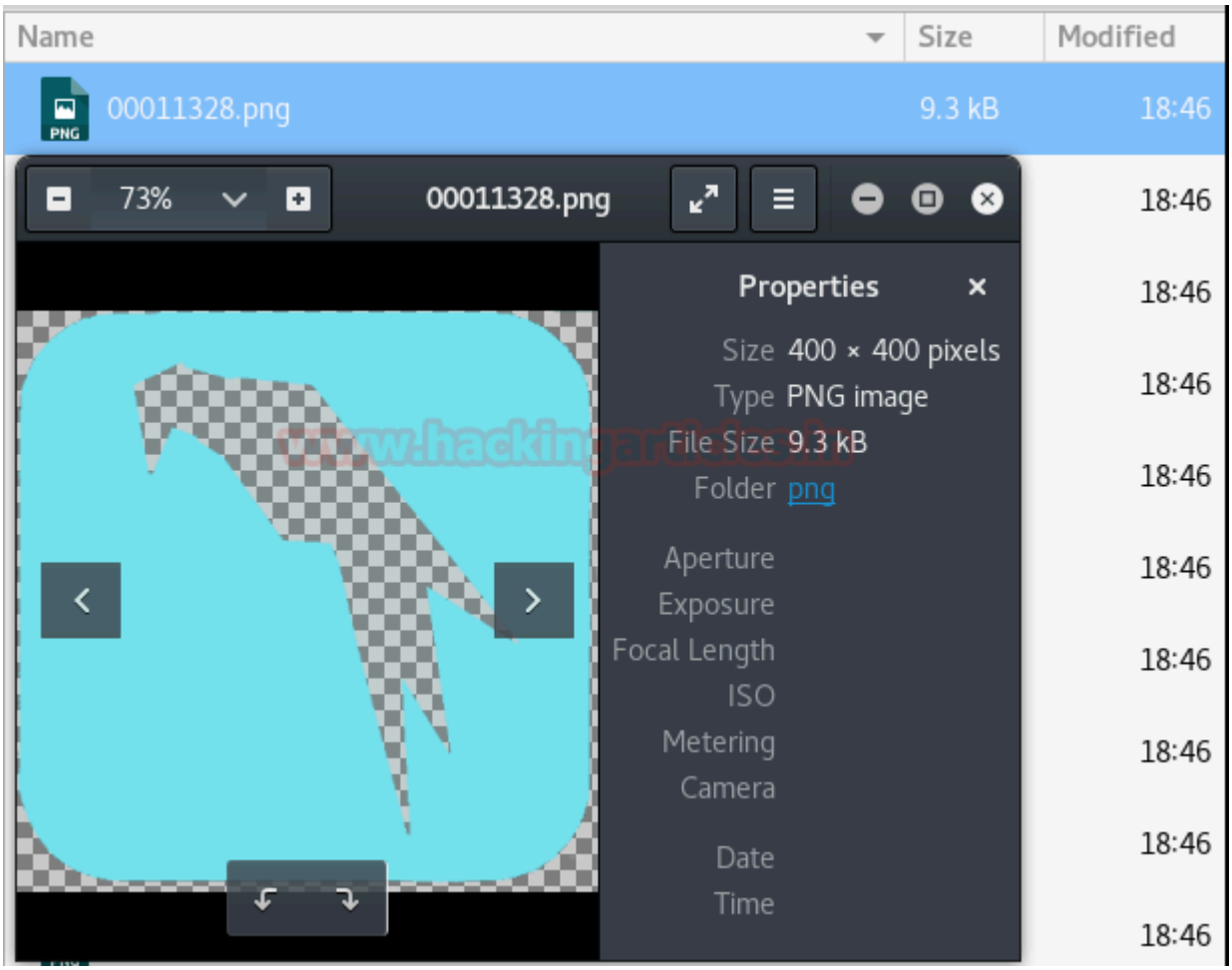
The end of the report contains shows the total files extracted with more particulars.

```
74:     00192344.pdf         140 KB        98480128
75:     00192632.pdf          23 KB        98627584
76:     00011904.avi           2 MB         6094848
77:     00022928.avi           2 MB        11739136
78:     00127136.avi           2 MB        65093632
79:     00159360.avi           2 MB        81592320
Finish: Fri Jan 12 19:46:00 2018

80 FILES EXTRACTED

jpg:= 4
png:= 61
zip:= 9
pdf:= 2
avi:= 4
```

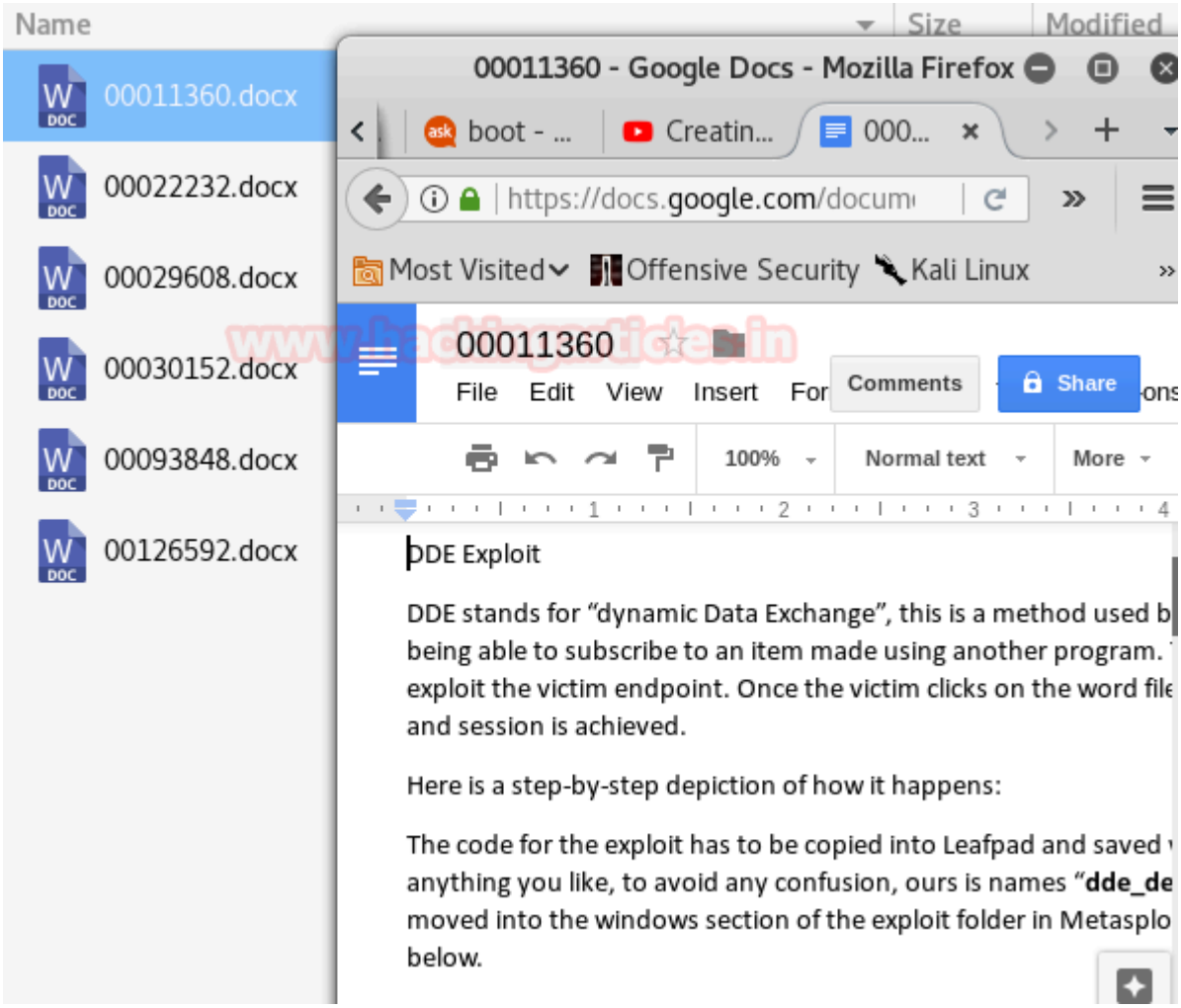

We will open one file from the **jpg** folder to see what we have.



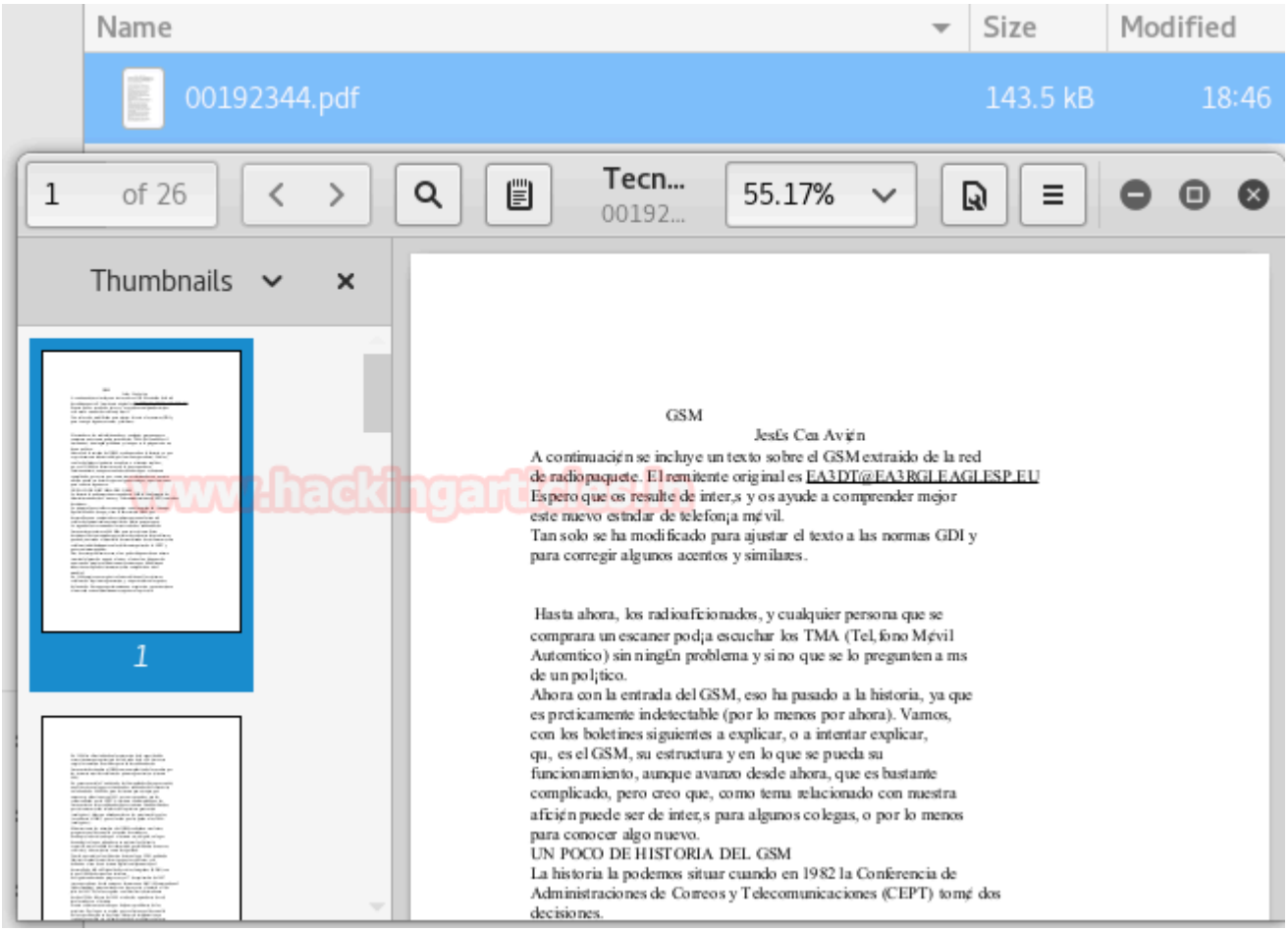
One from the **png** folder.



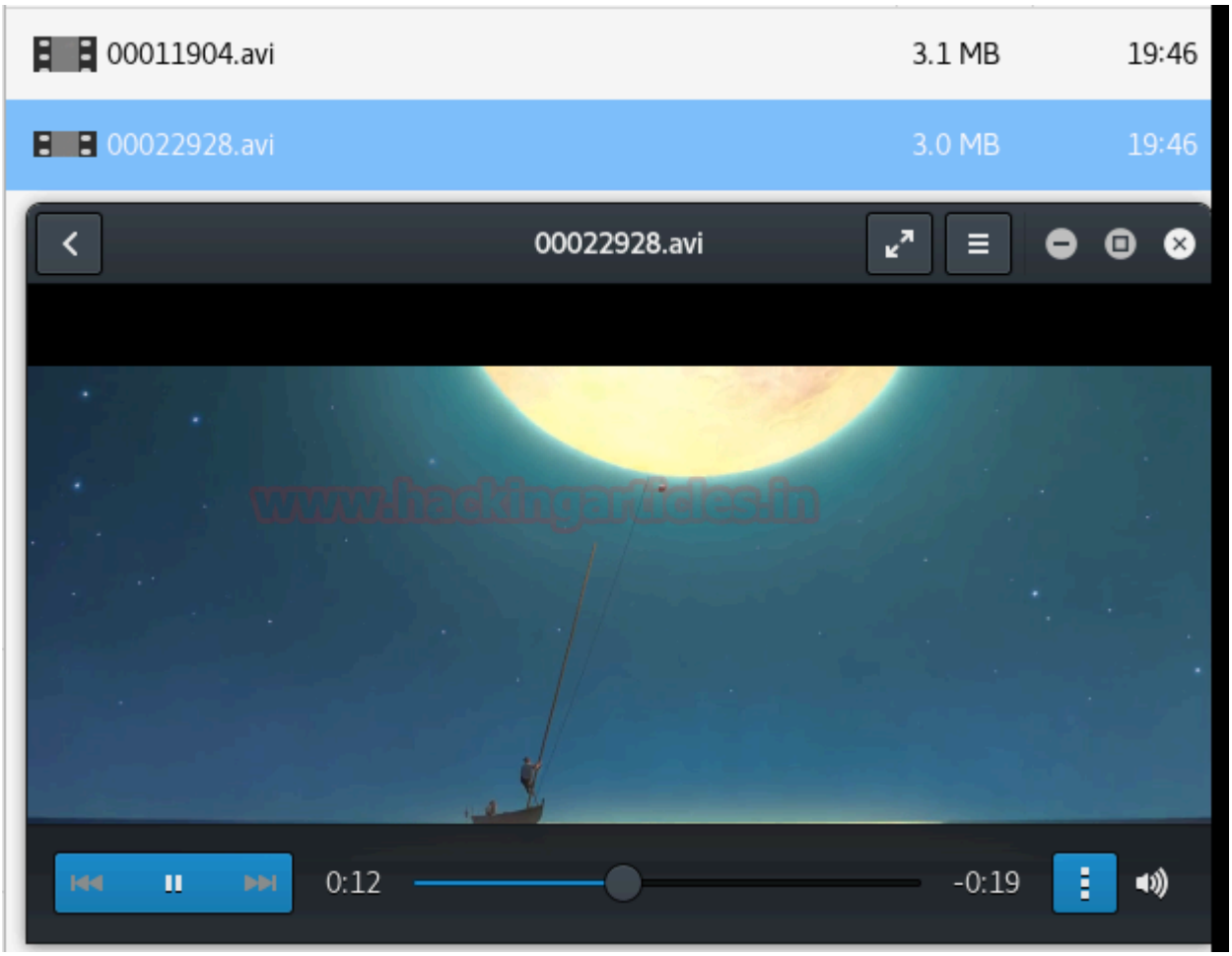
Inside the **docx** folder.



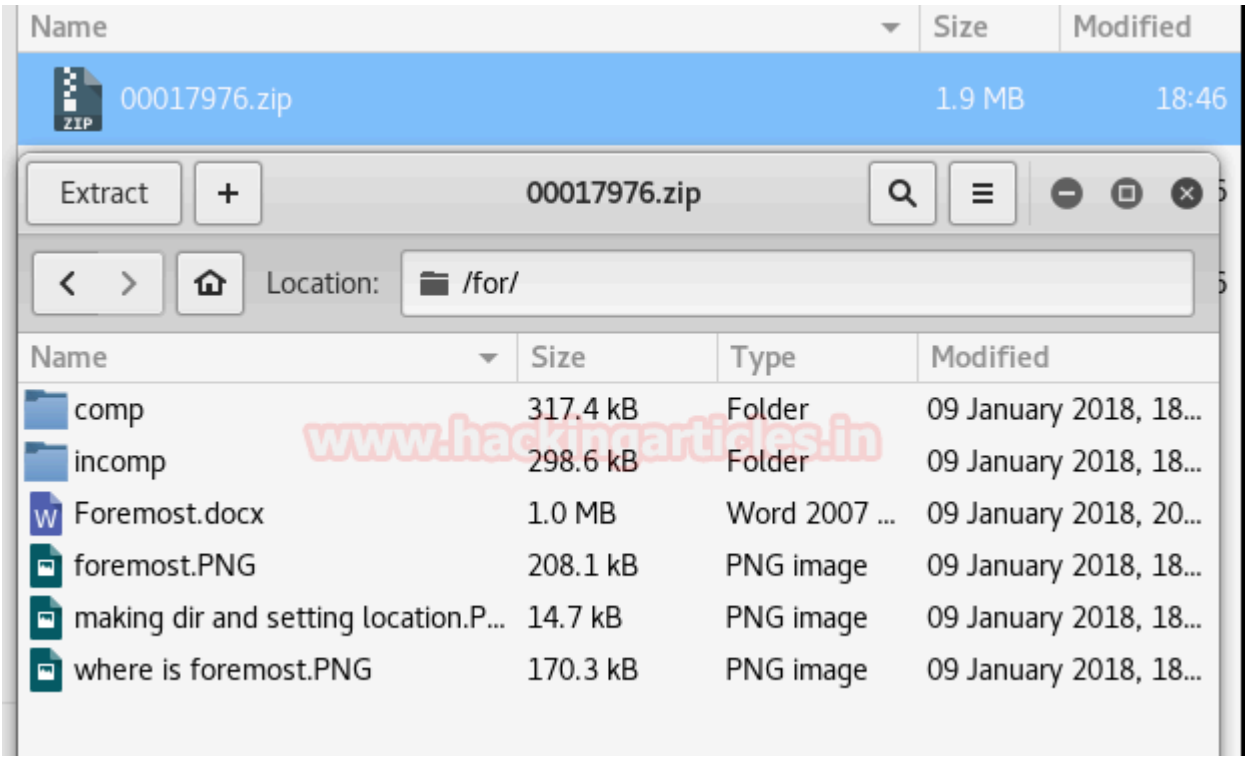
Inside the **pdf** folder.



Now the **avi** folder



And finally the **zip** folder.



As you can see, Foremost was successfully able to carve files out of the disk image file and give us the results. Let's put it to the test.

This a very interesting tool and its simplicity is what makes it stand out.

The only issue I could see with this is that the file names are not recovered, which can make the search process very tedious unless the option of automation and a frame of reference are available.

That being said, in forensics, just being able recover the files without opening or extracting disk image itself is a huge advantage, the reason for saying so is that, if you do extract or open the disk image you never know what might be waiting for you inside, this way you have more control over the entire investigation process. Enjoy using this tool.

Have fun and stay ethical.