Investigating Illegal Possession of Images

This case is all about illegal possession of Rhino images DFRWS 2005 RODEO CHALLENGE
NIST hosts the USB DD Image
BY : ROHAN KARTHIK

Http Analysis using Wireshark Basics (text)

Let us see how the analysis of wireshark is done for the case study we are doing

Configuring the wireshark

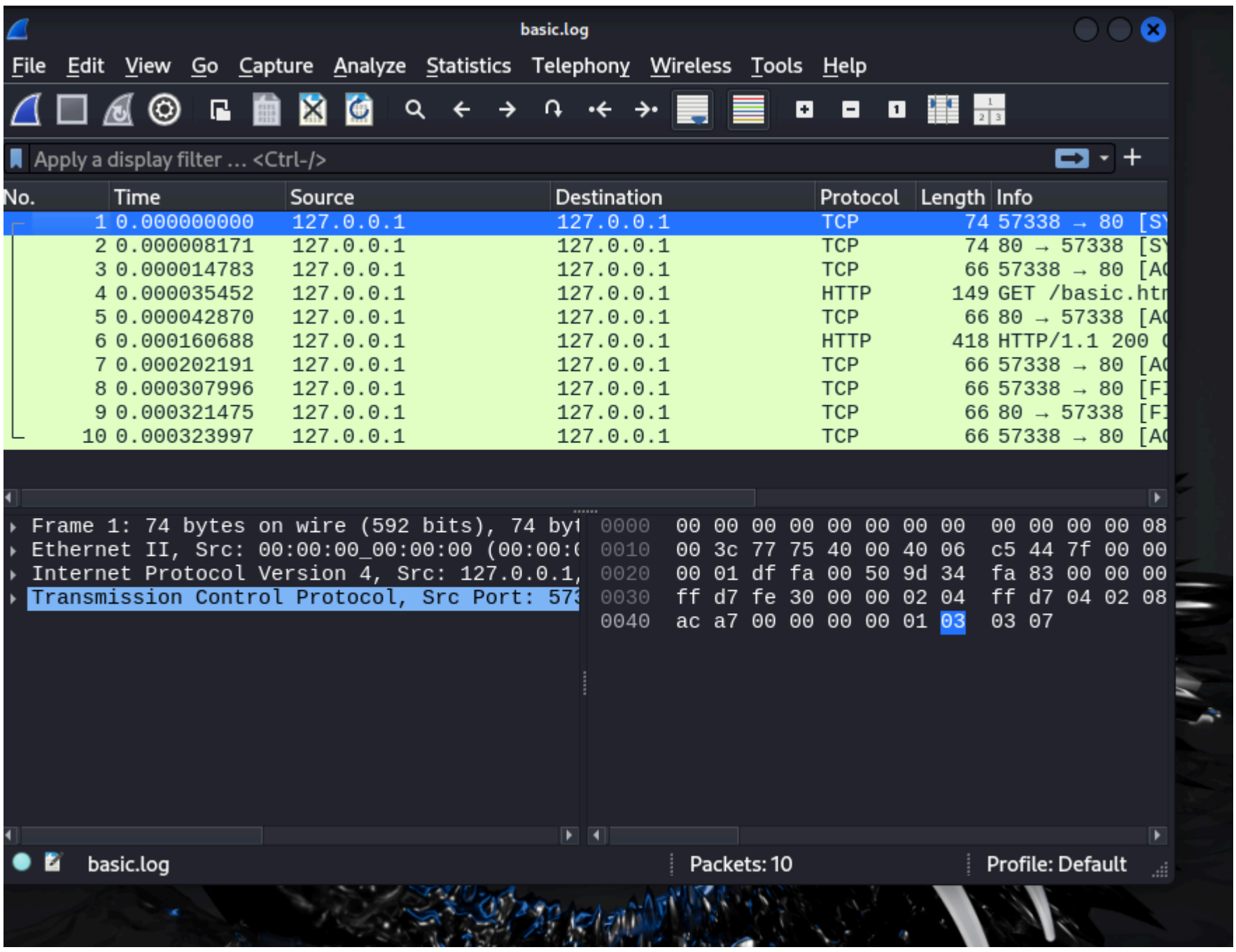
1.sudo dpkg-reconfigure wireshark-common
2.sudo usermod -a -G wireshark {your username} (The username which your kali is associated with)
3.logging out and logging back in again

Download the log file to analyze the evidence for the case by :

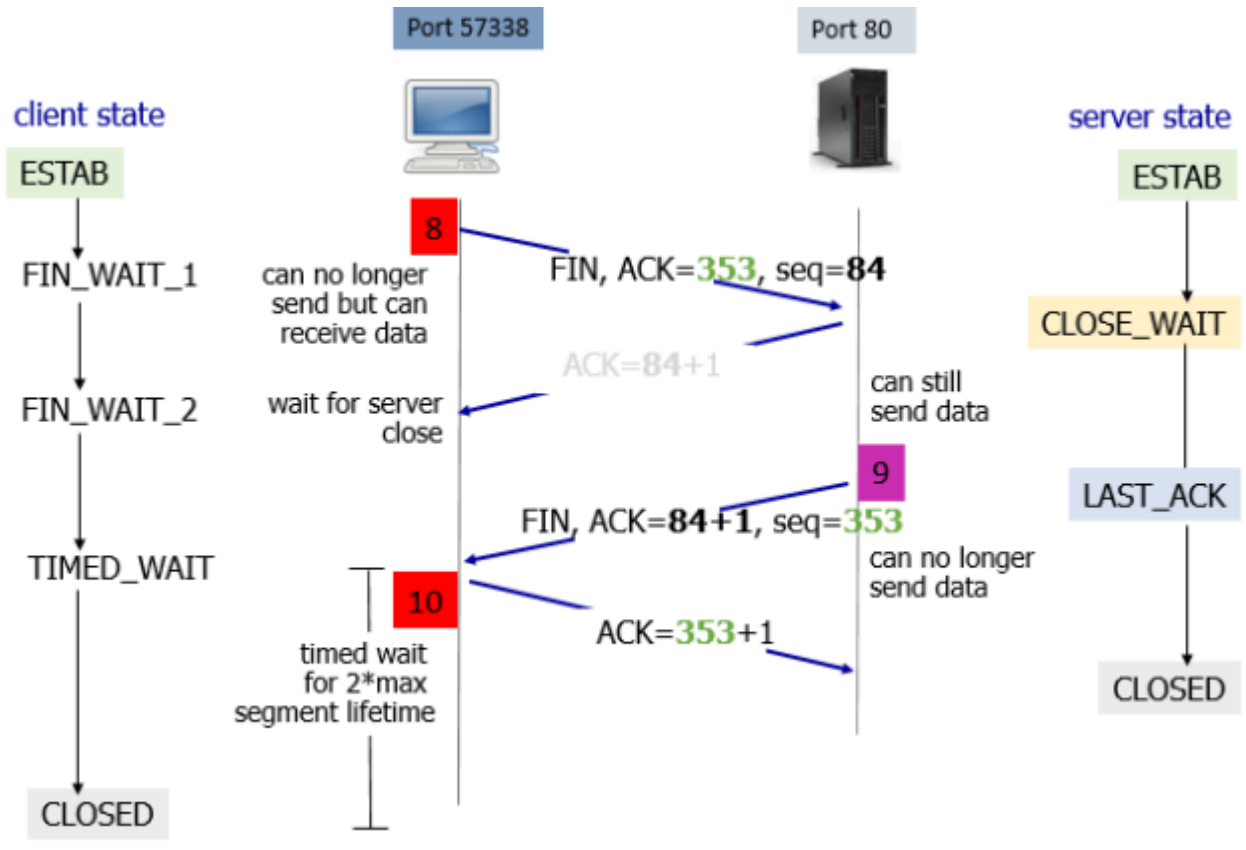
https://github.com/frankwxu/digital-forensics-lab/blob/main/Illegal_Possession_Images/lab_files/traffic/basic.log

Trying to Analyze the Log file

```
┌──(root㉿kali)-[/home/kali/rhino]
└─# cd traffic

┌──(root㉿kali)-[/home/kali/rhino/traffic]
└─# ls
basic.log

┌──(root㉿kali)-[/home/kali/rhino/traffic]
└─#
```



1. A TCP Three-way handshake and a Http Request and Response and a Tear down can be seen in the file of basic.log



Knowing about the Shred Tool

xxd :

It is the command which is used to create a hexadump of a file or standard input , it can reverse the process converting a hexadump back into its binary form

```
┌──(root㉿kali)-[/home/kali/Rhino_CaseStudy/shred]
└─# touch test.txt

┌──(root㉿kali)-[/home/kali/Rhino_CaseStudy/shred]
└─# nano test.txt
┌──(root㉿kali)-[/home/kali/Rhino_CaseStudy/shred]
└─# ls
test.txt

┌──(root㉿kali)-[/home/kali/Rhino_CaseStudy/shred]
└─# xxd test.txt
00000000: 4865 6c6c 6f0a                           Hello.

┌──(root㉿kali)-[/home/kali/Rhino_CaseStudy/shred]
└─#
```

```
┌──(root㉿kali)-[/home/kali/Rhino_CaseStudy/shred]
└─# xxd -b test.txt
00000000: 01001000 01100101 01101100 01101100 01101111 00001010  Hello.

┌──(root㉿kali)-[/home/kali/Rhino_CaseStudy/shred]
└─#
```

Shred

Randomly shift the content of the file three times and overwrite it with zero

shred -vzn 3 file

Overwrite the specified FILE(s) repeatedly, in order to make it harder for even very expensive hardware probing to recover the data.

```
┌──(root㉿kali)-[/home/kali/Rhino_CaseStudy/shred]
└─# shred -vzn 3 test.txt
shred: test.txt: pass 1/4 (random)...
shred: test.txt: pass 2/4 (random)...
shred: test.txt: pass 3/4 (random)...
shred: test.txt: pass 4/4 (000000)...
```

```
┌──(root㉿kali)-[/home/kali/Rhino_CaseStudy/shred]
└─# xxd test.txt
00000000: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000010: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000020: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000030: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000040: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000050: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000060: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000070: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000080: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000090: 0000 0000 0000 0000 0000 0000 0000 0000  ................
000000a0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
000000b0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
000000c0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
000000d0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
000000e0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
000000f0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000100: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000110: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000120: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000130: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000140: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000150: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000160: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000170: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000180: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000190: 0000 0000 0000 0000 0000 0000 0000 0000  ................
000001a0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
000001b0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
000001c0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
000001d0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
000001e0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
000001f0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000200: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000210: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000220: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000230: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000240: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000250: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000260: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000270: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000280: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000290: 0000 0000 0000 0000 0000 0000 0000 0000  ................
000002a0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
000002b0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
000002c0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
000002d0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
000002e0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
000002f0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000300: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000310: 0000 0000 0000 0000 0000 0000 0000 0000  ................
```

```
00000ce0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000cf0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000d00: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000d10: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000d20: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000d30: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000d40: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000d50: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000d60: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000d70: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000d80: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000d90: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000da0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000db0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000dc0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000dd0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000de0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000df0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000e00: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000e10: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000e20: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000e30: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000e40: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000e50: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000e60: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000e70: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000e80: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000e90: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000ea0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000eb0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000ec0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000ed0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000ee0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000ef0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000f00: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000f10: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000f20: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000f30: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000f40: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000f50: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000f60: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000f70: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000f80: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000f90: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000fa0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000fb0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000fc0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000fd0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000fe0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000ff0: 0000 0000 0000 0000 0000 0000 0000 0000  ................

┌──(root㉿kali)-[/home/kali/Rhino_CaseStudy/shred]
└─#
```

The size after the shreding is

```
┌──(root㉿kali)-[/home/kali/Rhino_CaseStudy/shred]
└─# ls -l test.txt
-rw-r--r-- 1 root root 4096 Mar 27 00:02 test.txt

┌──(root㉿kali)-[/home/kali/Rhino_CaseStudy/shred]
└─#
```

Let us see the example for an image

```
┌──(root㉿kali)-[/home/kali/Rhino_CaseStudy/shred]
└─# wget -q https://www.dropbox.com/s/p575df3fui42d88/660*450-city-shot.jpg

┌──(root㉿kali)-[/home/kali/Rhino_CaseStudy/shred]
└─# ls
'660*450-city-shot.jpg'   test.txt
```

Viewing the hex format of the image

```
┌──(root㉿kali)-[/home/kali/Rhino_CaseStudy/shred]
└─# xxd '660*450-city-shot.jpg'
00000000: ffd8 ffe1 227a 4578 6966 0000 4d4d 002a  ....."zExif..MM.*
00000010: 0000 0008 0007 0112 0003 0000 0001 0001  ................
00000020: 0000 011a 0005 0000 0001 0000 0062 011b  .............b..
00000030: 0005 0000 0001 0000 006a 0128 0003 0000  .........j.(....
00000040: 0001 0002 0000 0131 0002 0000 0022 0000  .......1.....".."
00000050: 0072 0132 0002 0000 0014 0000 0094 8769  .r.2...........i
00000060: 0004 0000 0001 0000 00a8 0000 00d4 000a  ................
00000070: fc80 0000 2710 000a fc80 0000 2710 4164  ....'.......'.Ad
00000080: 6f62 6520 5068 6f74 6f73 686f 7020 4343  obe Photoshop CC
00000090: 2032 3031 3820 2857 696e 646f 7773 2900   2018 (Windows).
000000a0: 3230 3138 3a30 383a 3331 2031 363a 3235  2018:08:31 16:25
000000b0: 3a30 3100 0003 a001 0003 0000 0001 ffff  :01.. ... . ... . ..
000000c0: 0000 a002 0004 0000 0001 0000 0294 a003  .. .. . ... . .. ....
000000d0: 0004 0000 0001 0000 01c2 0000 0000 0000  . ... . .. .. ......
000000e0: 0006 0103 0003 0000 0001 0006 0000 011a  . ... . . ... . . .. ..
000000f0: 0005 0000 0001 0000 0122 011b 0005 0000  . . ... . .. ." .. . ..
00000100: 0001 0000 012a 0128 0003 0000 0001 0002  . . .. . *.(. . ... . .
00000110: 0000 0201 0004 0000 0001 0000 0132 0202  .. .. . . ... . .. .2 ..
00000120: 0004 0000 0001 0000 2140 0000 0000 0000  . . ... . .. !@......
00000130: 0048 0000 0001 0000 0048 0000 0001 ffd8  .H... . ...H... . ..
```

```
┌──(root㉿kali)-[/home/kali/Rhino_CaseStudy/shred]
└─# xxd '660*450-city-shot.jpg' | head
00000000: ffd8 ffe1 227a 4578 6966 0000 4d4d 002a  ....."zExif..MM.*
00000010: 0000 0008 0007 0112 0003 0000 0001 0001  ................
00000020: 0000 011a 0005 0000 0001 0000 0062 011b  .............b..
00000030: 0005 0000 0001 0000 006a 0128 0003 0000  .........j.(....
00000040: 0001 0002 0000 0131 0002 0000 0022 0000  .......1.....".."
00000050: 0072 0132 0002 0000 0014 0000 0094 8769  .r.2...........i
00000060: 0004 0000 0001 0000 00a8 0000 00d4 000a  ................
00000070: fc80 0000 2710 000a fc80 0000 2710 4164  ....'.......'.Ad
00000080: 6f62 6520 5068 6f74 6f73 686f 7020 4343  obe Photoshop CC
00000090: 2032 3031 3820 2857 696e 646f 7773 2900   2018 (Windows).
```

Size of the file without shred

```
┌──(root㉿kali)-[/home/kali/Rhino_CaseStudy/shred]
└─# ls -l '660*450-city-shot.jpg'
-rw-r--r-- 1 root root 432216 Mar 27 00:10 '660*450-city-shot.jpg'
```

```
┌──(root㉿kali)-[/home/kali/Rhino_CaseStudy/shred]
└─# shred -vzn 3 '660*450-city-shot.jpg'

shred: '660*450-city-shot.jpg': pass 1/4 (random)...
shred: '660*450-city-shot.jpg': pass 2/4 (random)...
shred: '660*450-city-shot.jpg': pass 3/4 (random)...
shred: '660*450-city-shot.jpg': pass 4/4 (000000)...

┌──(root㉿kali)-[/home/kali/Rhino_CaseStudy/shred]
└─#
```

Size of the File after Shred

```
┌──(root㉿kali)-[/home/kali/Rhino_CaseStudy/shred]
└─# ls -l '660*450-city-shot.jpg'
-rw-r--r-- 1 root root 434176 Mar 27 00:23 '660*450-city-shot.jpg'
```

Hex file of the image after shredding it

```
┌──(root㉿kali)-[/home/kali/Rhino_CaseStudy/shred]
└─# xxd '660*450-city-shot.jpg' | head
00000000: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000010: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000020: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000030: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000040: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000050: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000060: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000070: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000080: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000090: 0000 0000 0000 0000 0000 0000 0000 0000  ................
```

Rhion_Possession_Case Intro

The Case Study DD Image

The Rhino Hunt data set requires examination of a small image file and three network traces.

This image was contributed by Dr. Golden G. Richard III, and was originally used in the DFRWS 2005 RODEO CHALLENGE.

Scenario:

The city of New Orleans passed a law in 2004 making possession of nine or more unique rhinoceros images a serious crime. The network administrator at the University of New Orleans recently alerted police when his instance of RHINOVORE flagged illegal rhino traffic. Evidence in the case includes a computer and USB key seized from one of the University's labs.

Unfortunately, the computer had no hard drive. The USB key was imaged and a copy of the *dd* image is on the CD-ROM you've been given.

In addition to the USB key drive image, three network traces are also available—these were provided by the network administrator and involve the machine with the missing hard drive. The suspect is the primary user of this machine, who has been pursuing his Ph.D. at the University since 1972.

MD5 hashes for evidence:

c0d0093eb1664cd7b73f3a5225ae3f30 *rhino.log*
cd21eaf4acfb50f71ffff857d7968341 rhino2.log
7e29f9d67346df25faaf18efcd95fc30 *rhino3.log*
80348c58eec4c328ef1f7709adc56a54 RHINOUSB.dd

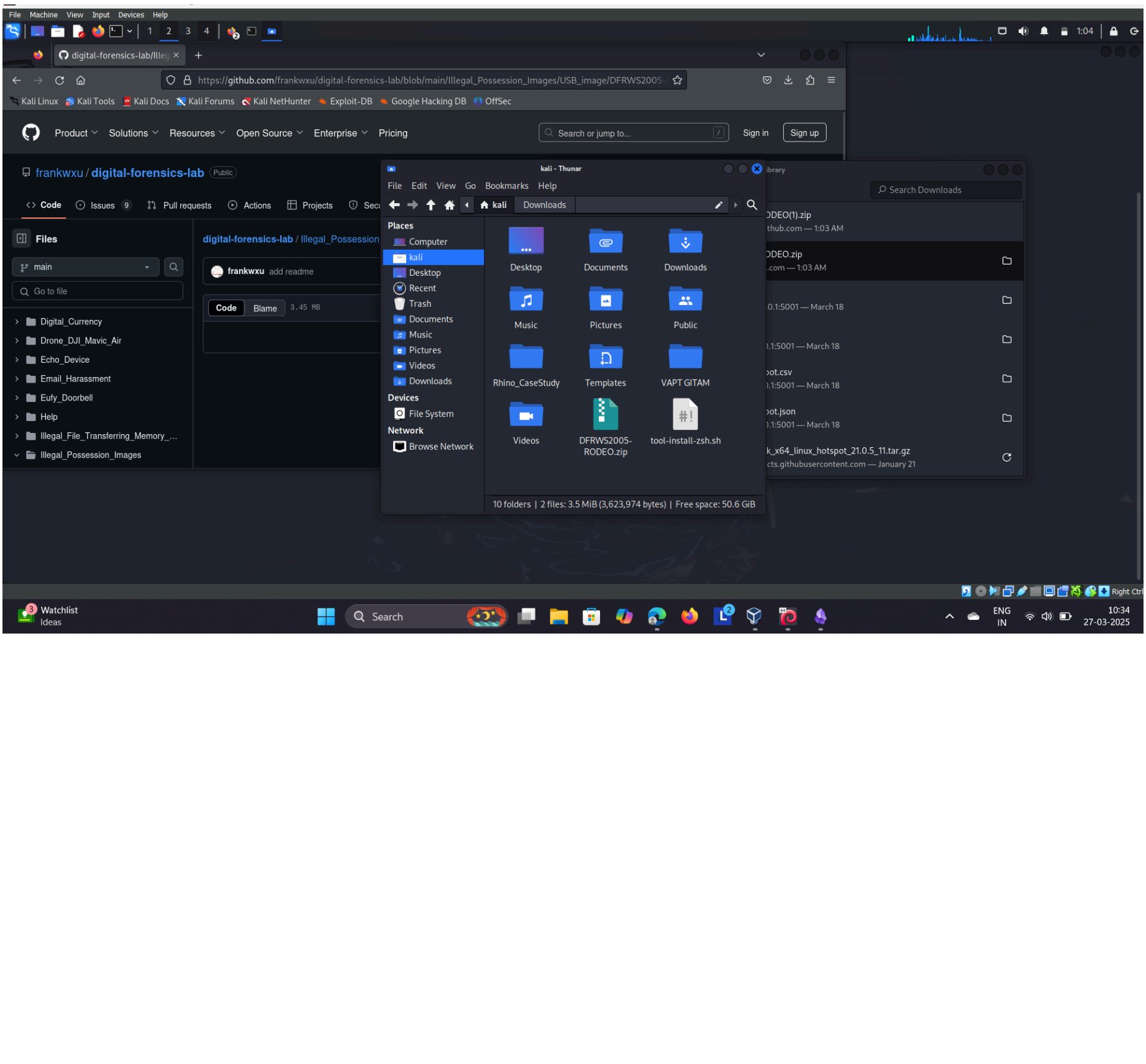
The image and trace files are in a zip archive.

The task:

Recover at least nine rhino pictures from the available evidence and include them in a brief report. In your report, provide answers to as many of the following questions as possible:

- Who gave the accused a telnet/ftp account?
- What's the username/password for the account?
- What relevant file transfers appear in the network traces?
- What happened to the hard drive in the computer? Where is it now?
- What happened to the USB key?
- What is recoverable from the *dd* image of the USB key?
- Is there any evidence that connects the USB key and the network traces? If so, what?

Extraction Phase 1 : Downloading the DD file and Recovering Images and any evidence Related



Creating a Image_File Directory and moving the downloaded Image file into that directory

```
┌──(root㉿kali)-[/home/kali/Rhino_CaseStudy/rhino]
└─# ls
Image_File

┌──(root㉿kali)-[/home/kali/Rhino_CaseStudy/rhino]
└─# cd ..

┌──(root㉿kali)-[/home/kali/Rhino_CaseStudy]
└─# cd ..

┌──(root㉿kali)-[/home/kali]
└─# ls
 Desktop              Documents   Music      Public           Templates             'VAPT GITAM'
 DFRWS2005-RODEO.zip  Downloads   Pictures   Rhino_CaseStudy  tool-install-zsh.sh    Videos

┌──(root㉿kali)-[/home/kali]
└─# mv DFRWS2005-RODEO.zip Rhino_CaseStudy/rhino/Image_File

┌──(root㉿kali)-[/home/kali]
└─#
```

After Downloading the image file we need to unzip it for further analysis

```
┌──(root㉿kali)-[/home/kali/Rhino_CaseStudy/rhino/Image_File]
└─# ls
DFRWS2005-RODEO.zip

┌──(root㉿kali)-[/home/kali/Rhino_CaseStudy/rhino/Image_File]
└─# unzip DFRWS2005-RODEO.zip
Archive:  DFRWS2005-RODEO.zip
  inflating: RHINOUSB.dd
  inflating: rhino.log
  inflating: rhino2.log
  inflating: rhino3.log

┌──(root㉿kali)-[/home/kali/Rhino_CaseStudy/rhino/Image_File]
└─#
```

Checking for the Hashes of the files

openssl dgst -md5 File_name

```
┌──(root㉿kali)-[/home/kali/Rhino_CaseStudy/rhino/Image_File]
└─# openssl dgst -md5 RHINOUSB.dd
MD5(RHINOUSB.dd)= 80348c58eec4c328ef1f7709adc56a54

┌──(root㉿kali)-[/home/kali/Rhino_CaseStudy/rhino/Image_File]
└─# openssl dgst -md5 rhino2.log
MD5(rhino2.log)= cd21eaf4acfb50f71ffff857d7968341

┌──(root㉿kali)-[/home/kali/Rhino_CaseStudy/rhino/Image_File]
└─# openssl dgst -md5 rhino3.log
MD5(rhino3.log)= 7e29f9d67346df25faaf18efcd95fc30

┌──(root㉿kali)-[/home/kali/Rhino_CaseStudy/rhino/Image_File]
└─# openssl dgst -md5 rhino.log
MD5(rhino.log)= c0d0093eb1664cd7b73f3a5225ae3f30

┌──(root㉿kali)-[/home/kali/Rhino_CaseStudy/rhino/Image_File]
└─#
```

Viewing the Partitions using fdisk

Fdisk command can view, create, delete change resize, copy and move partitions on hard drive

The `fdisk` command is a **partition management tool** used in Linux to create, delete, modify, and manage disk partitions. It is commonly used on **MBR (Master Boot Record)** partitioned disks.

```
┌──(root㉿kali)-[/home/kali/Rhino_CaseStudy/rhino/Image_File]
└─# fdisk -l
Disk /dev/sda: 80.09 GiB, 86000000000 bytes, 167968750 sectors
Disk model: VBOX HARDDISK
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0xc3a874f6

Device     Boot Start       End   Sectors  Size Id Type
/dev/sda1  *     2048 167968749 167966702 80.1G 83 Linux
```

```
┌──(root㉿kali)-[/home/kali/Rhino_CaseStudy/rhino/Image_File]
└─# fdisk -l RHINOUSB.dd
Disk RHINOUSB.dd: 247.48 MiB, 259506176 bytes, 506848 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
Disklabel type: dos
Disk identifier: 0x00000000
```

**Understanding the result

`Disk RHINOUSB.dd: 247.48 MiB, 259506176 bytes, 506848 sectors`

- `247.48 MiB` → The total size of the disk is **247.48 mebibytes** (MiB).
- `259506176 bytes` → The same size in **bytes**.
- `506848 sectors` → The total number of sectors on the disk.

Sector Details*

`Units: sectors of 1 * 512 = 512 bytes Sector size (logical/physical): 512 bytes / 512 bytes I/O size (minimum/optimal): 512 bytes / 512 bytes`

- Each **sector** is **512 bytes** (both logical & physical size).
- The **minimum and optimal I/O size** is also **512 bytes**.
- Meaning, this disk is formatted with a standard **512-byte sector size**, which is common for many storage devices.

Partition Table Type*

`Disklabel type: dos`

- This means the disk uses a **DOS (MBR - Master Boot Record) partition table**.
- MBR supports up to **4 primary partitions** or **3 primary + 1 extended partition**.
- It is commonly used in older systems but has limitations (max **2TB disk size**).

**Disk Identifier

`Disk identifier: 0x00000000`

- The **disk identifier** is a unique hex value assigned to the disk.
- Here, it is `0x00000000`, which is **unusual** because:
  - It suggests a **corrupted** or **blank disk**.
  - This may indicate a **raw disk image**, an **uninitialized disk**, or one created with certain forensic tools.

Possible Issues or Next Steps

1. **Check for Partitions**
   Run:

   bash

   CopyEdit

   `sudo fdisk -l RHINOUSB.dd`

   or

   bash

   CopyEdit

   `sudo parted RHINOUSB.dd print`

   to see if partitions exist.
2. **Check for Filesystem**
   Run:

   bash

   CopyEdit

   `sudo file -s RHINOUSB.dd`

   This will help determine if the disk contains a valid filesystem (e.g., FAT32, NTFS, ext4).

3. **Recover Partitions (if needed)**
   If partitions are missing or corrupted, use `testdisk`:

   bash

   CopyEdit

   ```
   sudo testdisk RHINOUSB.dd
   ```

Conclusion

- The disk image `RHINOUSB.dd` is **247.48 MiB**, formatted with an **MBR (DOS) partition table**.
- The **disk identifier is** `0x00000000`, which may indicate a raw or uninitialized disk.
- Further analysis with `fdisk`, `parted`, **or** `testdisk` can help determine if the disk contains valid data.

Further Analysis

mmls : media management "mmls" can show unallocated sectors so it can be used to search for hidden data

```
┌──(root㉿kali)-[/home/kali/Rhino_CaseStudy/rhino/Image_File]
└─# mmls RHINOUSB.dd

┌──(root㉿kali)-[/home/kali/Rhino_CaseStudy/rhino/Image_File]
└─# mmls RHINOUSB.dd
```

parted RHINOUSB.dd : to see whether the partition is available or not

```
┌──(root㉿kali)-[/home/kali/Rhino_CaseStudy/rhino/Image_File]
└─# parted RHINOUSB.dd
GNU Parted 3.6
Using /home/kali/Rhino_CaseStudy/rhino/Image_File/RHINOUSB.dd
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted) print
Model:  (file)
Disk /home/kali/Rhino_CaseStudy/rhino/Image_File/RHINOUSB.dd: 260MB
Sector size (logical/physical): 512B/512B
Partition Table: loop
Disk Flags:

Number  Start  End    Size   File system  Flags
 1      0.00B  260MB  260MB  fat16
```

fsstat : shows file system details and statistics including layout, sizes and labels
fsstat -b 512 -o 0 File_Name

```
┌──(root㉿kali)-[/home/kali/Rhino_CaseStudy/rhino/Image_File]
└─# fsstat -b 512 -o 0 RHINOUSB.dd
FILE SYSTEM INFORMATION
--------------------------------------------

File System Type: FAT16

OEM Name: mkdosfs
Volume ID: 0x4092d9d1
Volume Label (Boot Sector):
Volume Label (Root Directory):
File System Type Label: FAT16

Sectors before file system: 0

File System Layout (in sectors)
Total Range: 0 - 506847
* Reserved: 0 - 0
** Boot Sector: 0
* FAT 0: 1 - 248
* FAT 1: 249 - 496
* Data Area: 497 - 506847
** Root Directory: 497 - 528
** Cluster Area: 529 - 506840
** Non-clustered: 506841 - 506847

METADATA INFORMATION
--------------------------------------------

Range: 2 - 8101622
Root Directory: 2

CONTENT INFORMATION
```

```
CONTENT INFORMATION
--------------------------------------------

Sector Size: 512
Cluster Size: 4096
Total Cluster Range: 2 - 63290

FAT CONTENTS (in sectors)
--------------------------------------------

529-536 (8) -> EOF
537-544 (8) -> EOF
```

fls -o 0 file_name
For listing the files

```
┌──(root㉿kali)-[/home/kali/Rhino_CaseStudy/rhino/Image_File]
└─# fls -o 0 RHINOUSB.dd
r/r 4:  gumbo1.txt
r/r 6:  gumbo2.txt
v/v 8101619:    $MBR
v/v 8101620:    $FAT1
v/v 8101621:    $FAT2
V/V 8101622:    $OrphanFiles
```

```
┌──(root㉿kali)-[/home/kali/Rhino_CaseStudy/rhino/Image_File]
└─# fls -D RHINOUSB.dd
V/V 8101622:    $OrphanFiles

┌──(root㉿kali)-[/home/kali/Rhino_CaseStudy/rhino/Image_File]
└─# fls -r RHINOUSB.dd
r/r 4:  gumbo1.txt
r/r 6:  gumbo2.txt
v/v 8101619:    $MBR
v/v 8101620:    $FAT1
v/v 8101621:    $FAT2
V/V 8101622:    $OrphanFiles

┌──(root㉿kali)-[/home/kali/Rhino_CaseStudy/rhino/Image_File]
└─# fls -u  RHINOUSB.dd
r/r 4:  gumbo1.txt
r/r 6:  gumbo2.txt
v/v 8101619:    $MBR
v/v 8101620:    $FAT1
v/v 8101621:    $FAT2
V/V 8101622:    $OrphanFiles
```

```
┌──(root㉿kali)-[/home/kali/Rhino_CaseStudy/rhino/Image_File]
└─# fls
Missing image name
usage: fls [-adDFlhpruvV] [-f fstype] [-i imgtype] [-b dev_sector_size] [-m dir/] [-o imgoffset] [-z ZONE] [-s second
s] image [images] [inode]
        If [inode] is not given, the root directory is used
        -a: Display "." and ".." entries
        -d: Display deleted entries only
        -D: Display only directories
        -F: Display only files
        -l: Display long version (like ls -l)
        -i imgtype: Format of image file (use '-i list' for supported types)
        -b dev_sector_size: The size (in bytes) of the device sectors
        -f fstype: File system type (use '-f list' for supported types)
        -m: Display output in mactime input format with
              dir/ as the actual mount point of the image
        -h: Include MD5 checksum hash in mactime output
        -o imgoffset: Offset into image file (in sectors)
        -P pooltype: Pool container type (use '-P list' for supported types)
        -B pool_volume_block: Starting block (for pool volumes only)
        -S snap_id: Snapshot ID (for APFS only)
        -p: Display full path for each file
        -r: Recurse on directory entries
        -u: Display undeleted entries only
        -v: verbose output to stderr
        -V: Print version
        -z: Time zone of original machine (i.e. EST5EDT or GMT) (only useful with -l)
        -s seconds: Time skew of original machine (in seconds) (only useful with -l & -m)
        -k password: Decryption password for encrypted volumes
```

```
┌──(root㉿kali)-[/home/kali/Rhino_CaseStudy/rhino/Image_File]
└─# fls -d  RHINOUSB.dd

┌──(root㉿kali)-[/home/kali/Rhino_CaseStudy/rhino/Image_File]
└─#
```

We need to recover the files by copying the dd file to other name to avoid tampering the Evidence

```
┌──(root㉿kali)-[/home/kali/Rhino_CaseStudy/rhino/Image_File]
└─# cp RHINOUSB.dd RHINOUSB_My_Copy.dd

┌──(root㉿kali)-[/home/kali/Rhino_CaseStudy/rhino/Image_File]
└─# ls
DFRWS2005-RODEO.zip  rhino2.log  rhino3.log  rhino.log  RHINOUSB.dd  RHINOUSB_My_Copy.dd

┌──(root㉿kali)-[/home/kali/Rhino_CaseStudy/rhino/Image_File]
└─#
```

Let us recover the files in the image
icat -o 0 RHINOUSB_My_Copy.dd 4 > gumbo1_output.txt
icat -o 0 RHINOUSB_My_Copy.dd 6 > gumbo2_output.txt

```
┌──(root㉿kali)-[/home/kali/Rhino_CaseStudy/rhino/Image_File]
└─# icat -o 0 RHINOUSB_My_Copy.dd 4 > gumbo1_output.txt

┌──(root㉿kali)-[/home/kali/Rhino_CaseStudy/rhino/Image_File]
└─# icat -o 0 RHINOUSB_My_Copy.dd 6 > gumbo2_output.txt
```

```
┌──(root㉿kali)-[/home/kali/Rhino_CaseStudy/rhino/Image_File]
└─# ls -l
total 514016
-rw-rw-r-- 1 kali kali   3614418 Mar 27 01:03 DFRWS2005-RODEO.zip
-rw-r--r-- 1 root root      2815 Mar 27 03:19 gumbo1_output.txt
-rw-r--r-- 1 root root      1293 Mar 27 03:19 gumbo2_output.txt
-rw-r--r-- 1 root root    292604 Apr 28  2004 rhino2.log
-rw-r--r-- 1 root root    226094 Apr 28  2004 rhino3.log
-rw-r--r-- 1 root root   3187907 Apr 26  2004 rhino.log
-rw-r--r-- 1 root root 259506176 Apr 30  2004 RHINOUSB.dd
-rw-r--r-- 1 root root 259506176 Mar 27 03:16 RHINOUSB_My_Copy.dd
```

Verify the extracted file names

```
┌──(root㉿kali)-[/home/kali/Rhino_CaseStudy/rhino/Image_File]
└─# cat  gumbo1_output.txt
SHRIMP AND TASSO GUMBO

Associate Food Editor: Alexis Touchet
Father: Rodney Miller, Abbeville, LA

When I was a kid, our summer routine started with the opening day of
shrimp season ♦ my dad would get the boat ready and out we would go,
winding our way down the bayou until we reached Vermilion Bay. He
would check to make sure the trawl net was securely tied before
tossing it into the water, and would then let it be dragged down
slowly along with the trawl boards. When we painstakingly pulled the
net up onto the boat and untied the bag, it spilled out a catch of
shrimp, crabs, and fish. These fresh shrimp were the stars in Dad's
favorite gumbo recipe.

2 lb large shrimp in shell (21 to 25 per lb), peeled and shells reserved
14 cups water
1/4 cup vegetable oil
1/2 cup all-purpose flour
2 medium onions, chopped
2 celery ribs, chopped
1 large green bell pepper, chopped
1/2 lb fresh or frozen pork tasso* (Cajun-cured smoked meat), thawed if frozen, trimmed and cut into 1/4-inch pieces
1 1/2 teaspoons salt
1/2 teaspoon cayenne
10 oz fresh or frozen baby okra, thawed if frozen, trimmed, and cut into 1/4-inch-thick rounds (2 cups)
3/4 cup thinly sliced scallion greens
Accompaniments: cooked white rice; hot pepper sauce

Simmer shrimp shells and water, uncovered, in an 8-quart pot until
liquid is reduced to about 12 cups, 15 to 20 minutes, then pour
through a sieve set over a large bowl and discard shells.

Stir together oil and flour in a 10-inch heavy skillet (preferably
cast-iron) with a flat metal or wooden spatula, then cook over
moderately low heat (do not use a high-BTU burner), scraping back and
forth constantly (not stirring) until roux is the color of milk
chocolate, 30 to 45 minutes. (As roux cooks, it may be necessary to
lower the heat to prevent scorching.) Add onions, celery, and bell
pepper and cook, scraping back and forth occasionally, until onion is
softened, about 8 minutes.

Scrape roux mixture into cleaned 8-quart pot, then add shrimp stock and bring to a boil, stirring occasionally. Reduc
e heat, then add tasso, salt, and cayenne and simmer, uncovered, 30 minutes. Add okra and simmer until tender, 5 to 8
 minutes. Stir in shrimp and simmer until just cooked through, 2 to 3 minutes. Stir in scallion greens and salt to ta
ste.

Cooks' notes: Andouille or other smoked pork sausage can be
```

```
┌──(root㉿kali)-[/home/kali/Rhino_CaseStudy/rhino/Image_File]
└─# cat  gumbo2_output.txt
SHRIMP AND ANDOUILLE SAUSAGE GUMBO

1/2 cup vegetable oil
1/2 cup all purpose flour
4 celery stalks, coarsely chopped
2 medium onions, coarsely chopped
2 green bell peppers, chopped
2 bay leaves
2 teaspoons salt
2 teaspoons dried oregano, crumbled
1/2 teaspoon cayenne pepper
5 8-ounce bottles clam juice
1 28-ounce can plum tomatoes, drained, chopped
1 pound smoked andouille or kielbasa sausage, halved lengthwise,
 sliced 1/4 inch thick
1/2 pound okra, trimmed, cut crosswise into 1/4-inch-thick slices

2 pounds uncooked medium shrimp, peeled, deveined
Freshly cooked long-grain rice
2 tomatoes, seeded, diced

Heat oil heavy large Dutch oven over high heat until almost
smoking. Add flour and stir until dark red-brown, about 8
minutes. Immediately add celery, onions and bell peppers. Cook 5
minutes, stirring and scraping bottom of pan often. Mix in bay leaves,
salt, oregano and cayenne. Add clam juice, canned tomatoes and
sausage. Boil 15 minutes. Add okra, reduce heat and simmer until okra
is tender, about 15 minutes. (Can be made 1 day ahead. Cover and
chill. Bring to simmer before continuing.)

Add shrimp to gumbo and simmer until just cooked through, about 3
minutes. Mound rice in soup bowls. Ladle gumbo over. Sprinkle with
tomatoes.

Serves 8.
Bon App♦tit
November 1992
```

There is nothing much of use from this result .

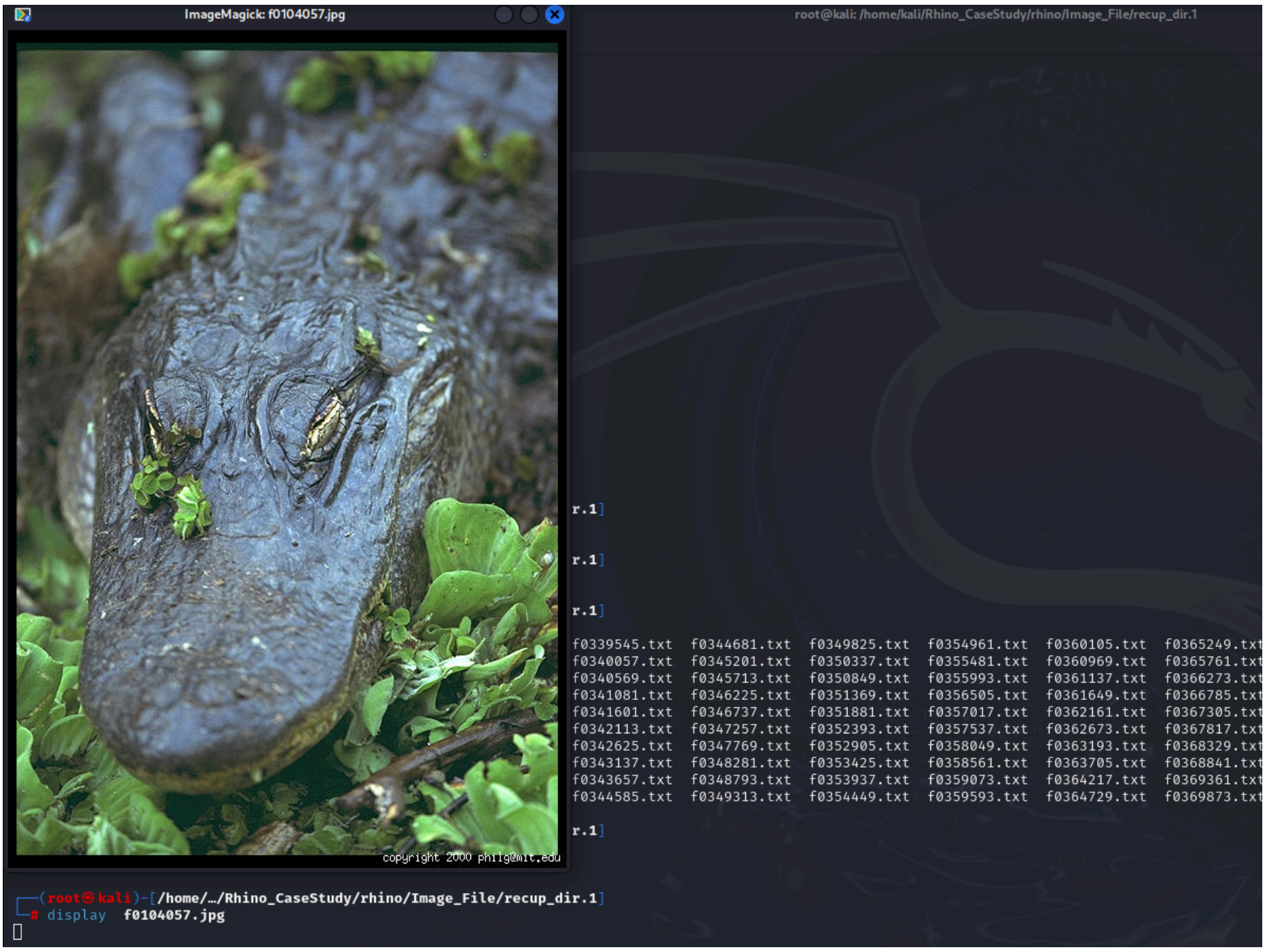
Verifying the hashes of both the Images

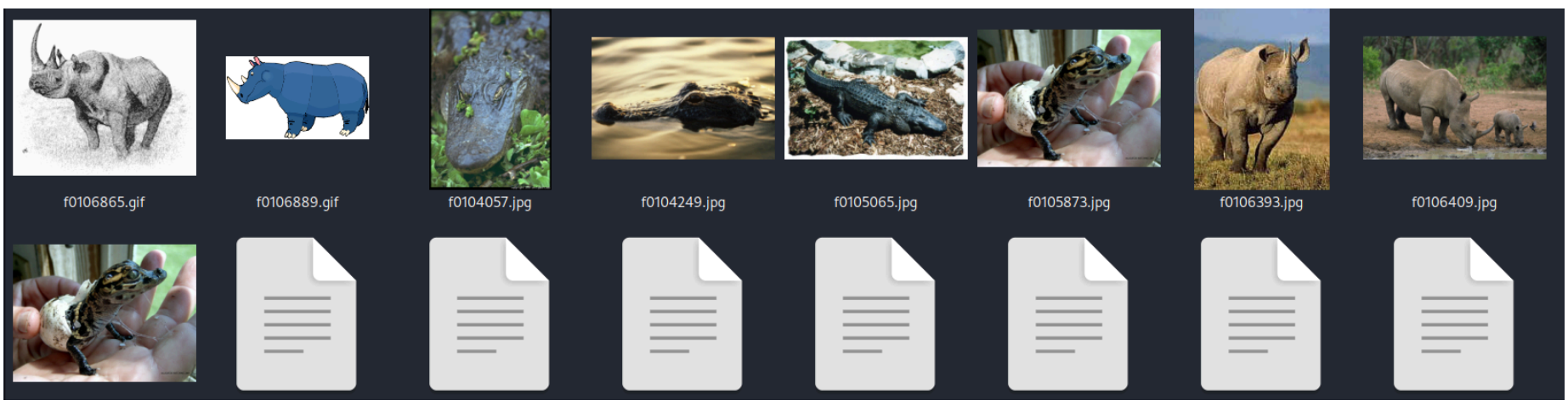
```
┌──(root㉿kali)-[/home/kali/Rhino_CaseStudy/rhino/Image_File]
└─# openssl dgst -md5 RHINOUSB.dd
MD5(RHINOUSB.dd)= 80348c58eec4c328ef1f7709adc56a54

┌──(root㉿kali)-[/home/kali/Rhino_CaseStudy/rhino/Image_File]
└─# openssl dgst -md5 RHINOUSB_My_Copy.dd
MD5(RHINOUSB_My_Copy.dd)= 80348c58eec4c328ef1f7709adc56a54
```

Recovering the Rhino Images from the image file

Using Photorec for recovering the photos

```
┌──(root㉿kali)-[/home/kali/Rhino_CaseStudy/rhino/Image_File]
└─# photorec  RHINOUSB_My_Copy.dd
PhotoRec 7.1, Data Recovery Utility, July 2019
Christophe GRENIER <grenier@cgsecurity.org>
https://www.cgsecurity.org

┌──(root㉿kali)-[/home/kali/Rhino_CaseStudy/rhino/Image_File]
└─# ls
DFRWS2005-RODEO.zip  gumbo2_output.txt  recup_dir.1  rhino3.log  RHINOUSB.dd
gumbo1_output.txt    photorec.se2       rhino2.log   rhino.log   RHINOUSB_My_Copy.dd

┌──(root㉿kali)-[/home/…/Rhino_CaseStudy/rhino/Image_File/recup_dir.1]
└─# ls
f0000545.txt                                    f0343137.txt  f0357017.txt  f0370897.txt  f0384777.txt
f0104057.jpg                                    f0343657.txt  f0357537.txt  f0371417.txt  f0385297.txt
f0104249.jpg                                    f0344585.txt  f0358049.txt  f0371929.txt  f0385809.txt
f0105065.jpg                                    f0344681.txt  f0358561.txt  f0372441.txt  f0386321.txt
f0105873.jpg                                    f0345201.txt  f0359073.txt  f0372953.txt  f0386833.txt
f0106393.jpg                                    f0345713.txt  f0359593.txt  f0373473.txt  f0387353.txt
f0106409.jpg                                    f0346225.txt  f0360105.txt  f0373985.txt  f0387865.txt
f0106865.gif                                    f0346737.txt  f0360969.txt  f0374497.txt  f0388377.txt
f0106889.gif                                    f0347257.txt  f0361137.txt  f0375009.txt  f0388889.txt
f0106905.txt                                    f0347769.txt  f0361649.txt  f0375529.txt  f0389409.txt
f0335017_She_died_in_February_at_the_age_of_74.doc  f0348281.txt  f0362161.txt  f0376041.txt  f0389921.txt
f0335081.jpg                                    f0348793.txt  f0362673.txt  f0376553.txt  f0390433.txt
f0335601.txt                                    f0349313.txt  f0363193.txt  f0377353.txt  f0390945.txt
f0335945.txt                                    f0349825.txt  f0363705.txt  f0377585.txt  f0391465.txt
f0336457.txt                                    f0350337.txt  f0364217.txt  f0378097.txt  f0391977.txt
f0336969.txt                                    f0350849.txt  f0364729.txt  f0378609.txt  f0392489.txt
f0337489.txt                                    f0351369.txt  f0365249.txt  f0379129.txt  f0393001.txt
f0338001.txt                                    f0351881.txt  f0365761.txt  f0379641.txt  f0393737.txt
f0338513.txt                                    f0352393.txt  f0366273.txt  f0380153.txt  f0394033.txt
f0339025.txt                                    f0352905.txt  f0366785.txt  f0380665.txt  f0394545.txt
f0339545.txt                                    f0353425.txt  f0367305.txt  f0381185.txt  f0395065.txt
f0340057.txt                                    f0353937.txt  f0367817.txt  f0381697.txt  f0395577.txt
f0340569.txt                                    f0354449.txt  f0368329.txt  f0382209.txt  f0396089.txt
f0341081.txt                                    f0354961.txt  f0368841.txt  f0382721.txt  f0396601.txt
f0341601.txt                                    f0355481.txt  f0369361.txt  f0383241.txt  report.xml
f0342113.txt                                    f0355993.txt  f0369873.txt  f0383753.txt
f0342625.txt                                    f0356505.txt  f0370385.txt  f0384265.txt

┌──(root㉿kali)-[/home/…/Rhino_CaseStudy/rhino/Image_File/recup_dir.1]
└─# ls -l *.jpg
-rw-r--r-- 1 root root  95814 Mar 27 03:28 f0104057.jpg
-rw-r--r-- 1 root root 415534 Mar 27 03:28 f0104249.jpg
-rw-r--r-- 1 root root 411361 Mar 27 03:28 f0105065.jpg
-rw-r--r-- 1 root root 264600 Mar 27 03:28 f0105873.jpg
-rw-r--r-- 1 root root   6809 Mar 27 03:28 f0106393.jpg
-rw-r--r-- 1 root root 230665 Mar 27 03:28 f0106409.jpg
-rw-r--r-- 1 root root 264600 Mar 27 03:28 f0335081.jpg

┌──(root㉿kali)-[/home/…/Rhino_CaseStudy/rhino/Image_File/recup_dir.1]
```





Examination Phase 1 : Rhino_Stegnography

Extracting the hidden photos

Detecting which tool is used for hiding

Stegdetect

it analyses image files for steganographic content

it runs statistical tests to determine if steganoraphic content is present

it tries to find the system that has been used to embed the hidden info

it testes if information has been embedded or the info is added at the end of file

Created a Directory of Steg and copied the images and gifs of rhino to the steg directory

f0106409.jpg

f0106393.jpg

f0106865.gif

f0106889.gif

```
└─# mkdir steg

┌──(root㉿kali)-[/home/kali/Rhino_CaseStudy/rhino]
└─# ls
Image_File  steg

┌──(root㉿kali)-[/home/kali/Rhino_CaseStudy/rhino]
└─# cd Image_File/recup_dir.1

┌──(root㉿kali)-[/home/…/Rhino_CaseStudy/rhino/Image_File/recup_dir.1]
└─# mv /home/kali/Rhino_CaseStudy/rhino/Image_File/recup_dir.1/f0106409.jpg  /home/kali/Rhino_CaseStudy/rhino/steg

┌──(root㉿kali)-[/home/…/Rhino_CaseStudy/rhino/Image_File/recup_dir.1]
└─# mv /home/kali/Rhino_CaseStudy/rhino/Image_File/recup_dir.1/f0106393.jpg  /home/kali/Rhino_CaseStudy/rhino/steg

┌──(root㉿kali)-[/home/…/Rhino_CaseStudy/rhino/Image_File/recup_dir.1]
└─# ls
f0000545.txt                                  f0344585.txt  f0358049.txt  f0371929.txt  f0385809.txt
f0104057.jpg                                  f0344681.txt  f0358561.txt  f0372441.txt  f0386321.txt
f0104249.jpg                                  f0345201.txt  f0359073.txt  f0372953.txt  f0386833.txt
f0105065.jpg                                  f0345713.txt  f0359593.txt  f0373473.txt  f0387353.txt
f0105873.jpg                                  f0346225.txt  f0360105.txt  f0373985.txt  f0387865.txt
f0106865.gif                                  f0346737.txt  f0360969.txt  f0374497.txt  f0388377.txt
f0106889.gif                                  f0347257.txt  f0361137.txt  f0375009.txt  f0388889.txt
f0106905.txt                                  f0347769.txt  f0361649.txt  f0375529.txt  f0389409.txt
f0335017_She_died_in_February_at_the_age_of_74.doc  f0348281.txt  f0362161.txt  f0376041.txt  f0389921.txt
f0335081.jpg                                  f0348793.txt  f0362673.txt  f0376553.txt  f0390433.txt
f0335601.txt                                  f0349313.txt  f0363193.txt  f0377353.txt  f0390945.txt
f0335945.txt                                  f0349825.txt  f0363705.txt  f0377585.txt  f0391465.txt
f0336457.txt                                  f0350337.txt  f0364217.txt  f0378097.txt  f0391977.txt
f0336969.txt                                  f0350849.txt  f0364729.txt  f0378609.txt  f0392489.txt
f0337489.txt                                  f0351369.txt  f0365249.txt  f0379129.txt  f0393001.txt
f0338001.txt                                  f0351881.txt  f0365761.txt  f0379641.txt  f0393737.txt
f0338513.txt                                  f0352393.txt  f0366273.txt  f0380153.txt  f0394033.txt
f0339025.txt                                  f0352905.txt  f0366785.txt  f0380665.txt  f0394545.txt
f0339545.txt                                  f0353425.txt  f0367305.txt  f0381185.txt  f0395065.txt
f0340057.txt                                  f0353937.txt  f0367817.txt  f0381697.txt  f0395577.txt
f0340569.txt                                  f0354449.txt  f0368329.txt  f0382209.txt  f0396089.txt
f0341081.txt                                  f0354961.txt  f0368841.txt  f0382721.txt  f0396601.txt
f0341601.txt                                  f0355481.txt  f0369361.txt  f0383241.txt  report.xml
f0342113.txt                                  f0355993.txt  f0369873.txt  f0383753.txt
f0342625.txt                                  f0356505.txt  f0370385.txt  f0384265.txt
f0343137.txt                                  f0357017.txt  f0370897.txt  f0384777.txt
f0343657.txt                                  f0357537.txt  f0371417.txt  f0385297.txt

┌──(root㉿kali)-[/home/…/Rhino_CaseStudy/rhino/Image_File/recup_dir.1]
└─# mv /home/kali/Rhino_CaseStudy/rhino/Image_File/recup_dir.1/f0106865.gif  /home/kali/Rhino_CaseStudy/rhino/steg

┌──(root㉿kali)-[/home/…/Rhino_CaseStudy/rhino/Image_File/recup_dir.1]
└─# mv /home/kali/Rhino_CaseStudy/rhino/Image_File/recup_dir.1/f0106889.gif  /home/kali/Rhino_CaseStudy/rhino/steg
```

```
┌──(root㉿kali)-[/home/kali/Rhino_CaseStudy/rhino]
└─# ls
Image_File  steg

┌──(root㉿kali)-[/home/kali/Rhino_CaseStudy/rhino]
└─# cd steg

┌──(root㉿kali)-[/home/kali/Rhino_CaseStudy/rhino/steg]
└─# ls
f0106393.jpg  f0106409.jpg  f0106865.gif  f0106889.gif

┌──(root㉿kali)-[/home/kali/Rhino_CaseStudy/rhino/steg]
└─#
```

https://view.officeapps.live.com/op/view.aspx?src=https%3A%2F%2Fraw.githubusercontent.com%2Ffrankwxu%2Fdigital-forensics-lab%2Frefs%2Fheads%2Fmain%2FIllegal_Pc

We've opened your file for quick and easy viewing right in Microsoft Edge. Choose Download file if you want to use it later. Download file

f0335017_She_died_in_February_at_the_age_of_74

Do you have to be a gold member to put in background pics??

A little background: When I was 14, I had eye surgery to correct a birth defect. When I called them the other day to find out when they were open, I got someone very, very stern. And they sent a snotty fool down from Buffalo to run the store. However, after a while of dealing with her crap, management decided they wanted some more room in the store to put...whatever. What's the point.

Most of the rides we wanted to take were sold out, but we got to ride on a tall ship from 3-5, which is exactly what we wanted. I found this site that is full of surveys through some people who are now obsessed with the site.

Rhino pictures illegal? Makes me sick. I "hid" the photos…hehehehe. Apparently, if there are less than 10 photos, it's no big deal.

OK. Things are getting a little weird. I zapped the hard drive and then threw it into the Mississippi River. I'm gonna reformat my USB key after this entry, but try not to destroy the good stuff. I need to change the password on the gnome account that Jeremy gave me. I can probably just do that at Radio Shack.

Examination of Recovered Rhino Photos

2: jpg
2: gif

Check for any suspicious data in photo metadata

```
┌──(root㉿kali)-[/home/kali/Rhino_CaseStudy/rhino/steg]
└─# exiftool f0106409.jpg
ExifTool Version Number         : 12.76
File Name                       : f0106409.jpg
Directory                       : .
File Size                       : 231 kB
File Modification Date/Time     : 2025:03:27 03:28:53-04:00
File Access Date/Time           : 2025:03:27 05:04:50-04:00
File Inode Change Date/Time     : 2025:03:27 04:58:12-04:00
File Permissions                : -rw-r--r--
File Type                       : JPEG
File Type Extension             : jpg
MIME Type                       : image/jpeg
JFIF Version                    : 1.01
Resolution Unit                 : inches
X Resolution                    : 170
Y Resolution                    : 170
Image Width                     : 1024
Image Height                    : 685
Encoding Process                : Baseline DCT, Huffman coding
Bits Per Sample                 : 8
Color Components                : 3
Y Cb Cr Sub Sampling            : YCbCr4:2:0 (2 2)
Image Size                      : 1024×685
Megapixels                      : 0.701
```

```
┌──(root㉿kali)-[/home/kali/Rhino_CaseStudy/rhino/steg]
└─# exiftool f0106393.jpg
ExifTool Version Number         : 12.76
File Name                       : f0106393.jpg
Directory                       : .
File Size                       : 6.8 kB
File Modification Date/Time     : 2025:03:27 03:28:53-04:00
File Access Date/Time           : 2025:03:27 05:04:10-04:00
File Inode Change Date/Time     : 2025:03:27 04:58:34-04:00
File Permissions                : -rw-r--r--
File Type                       : JPEG
File Type Extension             : jpg
MIME Type                       : image/jpeg
JFIF Version                    : 1.00
Resolution Unit                 : inches
X Resolution                    : 72
Y Resolution                    : 72
Comment                         : LEAD Technologies Inc. V1.01
Image Width                     : 169
Image Height                    : 228
Encoding Process                : Baseline DCT, Huffman coding
Bits Per Sample                 : 8
Color Components                : 3
Y Cb Cr Sub Sampling            : YCbCr4:4:4 (1 1)
Image Size                      : 169×228
Megapixels                      : 0.039
```

```
┌──(root㉿kali)-[/home/kali/Rhino_CaseStudy/rhino/steg]
└─# exiftool f0106865.gif
ExifTool Version Number         : 12.76
File Name                       : f0106865.gif
Directory                       : .
File Size                       : 11 kB
File Modification Date/Time     : 2025:03:27 03:28:53-04:00
File Access Date/Time           : 2025:03:27 05:06:19-04:00
File Inode Change Date/Time     : 2025:03:27 04:59:24-04:00
File Permissions                : -rw-r--r--
File Type                       : GIF
File Type Extension             : gif
MIME Type                       : image/gif
GIF Version                     : 89a
Image Width                     : 290
Image Height                    : 246
Has Color Map                   : Yes
Color Resolution Depth          : 3
Bits Per Pixel                  : 3
Background Color                : 0
Image Size                      : 290×246
Megapixels                      : 0.071
```

```
┌──(root㉿kali)-[/home/kali/Rhino_CaseStudy/rhino/steg]
└─# exiftool f0106889.gif
ExifTool Version Number         : 12.76
File Name                       : f0106889.gif
Directory                       : .
File Size                       : 4.1 kB
File Modification Date/Time     : 2025:03:27 03:28:53-04:00
File Access Date/Time           : 2025:03:27 05:06:33-04:00
File Inode Change Date/Time     : 2025:03:27 04:59:31-04:00
File Permissions                : -rw-r--r--
File Type                       : GIF
File Type Extension             : gif
MIME Type                       : image/gif
GIF Version                     : 89a
Image Width                     : 150
Image Height                    : 87
Has Color Map                   : Yes
Color Resolution Depth          : 8
Bits Per Pixel                  : 8
Background Color                : 0
Image Size                      : 150×87
Megapixels                      : 0.013
```

Examination of Other Photos from the .DD for stegnography detection

Creating a directory called "steg" for stegnography detection and further cracking of files and copying the other images to the directory for examination

```
┌──(root㉿kali)-[/home/kali/Rhino_CaseStudy/rhino]
└─# ls
Image_File  steg

┌──(root㉿kali)-[/home/kali/Rhino_CaseStudy/rhino]
└─# cd steg

┌──(root㉿kali)-[/home/kali/Rhino_CaseStudy/rhino/steg]
└─# ls
f0104057.jpg  f0104249.jpg  f0105065.jpg  f0105873.jpg  f0335081.jpg
```

Usage of Stegdetect

stegdetect is the tool used for detection of stegnography within the image files in format of jpg,gif etc.

•It analyses image files for steganographic content.

•It runs statistical tests to determine if steganographic content is present,

•It tries to find the system that has been used to embed the hidden information.

•It tests if information has been embedded with

•Jsteg, outguess, jphide, invisible secrets, F5

•It tests if information has been added at the end of file
Camouflage
appendX

Before checking with stegdetect need to see with exiftool for the meta data of the images might be possible we can get some information form it


```
File Name                       : f0104057.jpg
Directory                       : .
File Size                       : 96 kB
File Modification Date/Time     : 2025:03:28 12:44:11-04:00
File Access Date/Time           : 2025:04:01 05:34:22-04:00
File Inode Change Date/Time     : 2025:03:28 12:44:11-04:00
File Permissions                : -rw-r--r--
File Type                       : JPEG
File Type Extension             : jpg
MIME Type                       : image/jpeg
JFIF Version                    : 1.01
Resolution Unit                 : None
X Resolution                    : 0
Y Resolution                    : 0
Comment                         : copyright 2000 philg@mit.edu
Image Width                     : 528
Image Height                    : 792
Encoding Process                : Baseline DCT, Huffman coding
Bits Per Sample                 : 8
Color Components                : 3
Y Cb Cr Sub Sampling            : YCbCr4:4:4 (1 1)
Image Size                      : 528×792
Megapixels                      : 0.418
```

```
root@kali: /home/kali/Rhino_CaseStudy/rhino/steg
File  Actions  Edit  View  Help
ExifTool Version Number         : 12.76
File Name                       : f0104249.jpg
Directory                       : .
File Size                       : 416 kB
File Modification Date/Time     : 2025:03:28 12:44:29-04:00
File Access Date/Time           : 2025:03:31 08:11:59-04:00
File Inode Change Date/Time     : 2025:03:28 12:44:29-04:00
File Permissions                : -rw-r--r--
File Type                       : JPEG
File Type Extension             : jpg
MIME Type                       : image/jpeg
JFIF Version                    : 1.02
Resolution Unit                 : None
X Resolution                    : 100
Y Resolution                    : 100
Image Width                     : 1349
Image Height                    : 900
Encoding Process                : Baseline DCT, Huffman coding
Bits Per Sample                 : 8
Color Components                : 3
Y Cb Cr Sub Sampling            : YCbCr4:4:4 (1 1)
Image Size                      : 1349×900
Megapixels                      : 1.2
```

```
└─# exiftool f0105065.jpg
ExifTool Version Number         : 12.76
File Name                       : f0105065.jpg
Directory                       : .
File Size                       : 411 kB
File Modification Date/Time     : 2025:03:28 12:44:45-04:00
File Access Date/Time           : 2025:03:28 12:45:43-04:00
File Inode Change Date/Time     : 2025:03:28 12:44:45-04:00
File Permissions                : -rw-r--r--
File Type                       : JPEG
File Type Extension             : jpg
MIME Type                       : image/jpeg
JFIF Version                    : 1.01
Resolution Unit                 : None
X Resolution                    : 1
Y Resolution                    : 1
Image Width                     : 1686
Image Height                    : 1122
Encoding Process                : Baseline DCT, Huffman coding
Bits Per Sample                 : 8
Color Components                : 3
Y Cb Cr Sub Sampling            : YCbCr4:2:0 (2 2)
Image Size                      : 1686×1122
Megapixels                      : 1.9
```

```
ExifTool Version Number         : 12.76
File Name                       : f0105873.jpg
Directory                       : .
File Size                       : 265 kB
File Modification Date/Time     : 2025:03:28 12:45:02-04:00
File Access Date/Time           : 2025:03:28 12:45:43-04:00
File Inode Change Date/Time     : 2025:03:28 12:45:02-04:00
File Permissions                : -rw-r--r--
File Type                       : JPEG
File Type Extension             : jpg
MIME Type                       : image/jpeg
JFIF Version                    : 1.02
Resolution Unit                 : None
X Resolution                    : 100
Y Resolution                    : 100
Quality                         : 79%
DCT Encode Version              : 100
APP14 Flags 0                   : [14], Encoded with Blend=1 downsampling
APP14 Flags 1                   : (none)
Color Transform                 : YCbCr
Image Width                     : 1024
Image Height                    : 768
Encoding Process                : Baseline DCT, Huffman coding
Bits Per Sample                 : 8
Color Components                : 3
Y Cb Cr Sub Sampling            : YCbCr4:4:4 (1 1)
Image Size                      : 1024×768
Megapixels                      : 0.786
```

we need to download the stegdetect by the github link :
git clone https://github.com/poizan42/stegdetect

```
┌──(root㉿kali)-[/home/kali/Rhino_CaseStudy/rhino/steg]
└─# stegdetect
^C

┌──(root㉿kali)-[/home/kali/Rhino_CaseStudy/rhino/steg]
└─# stegdetect --help
stegdetect: invalid option -- '-'
Usage: stegdetect [-nqV] [-s <float>] [-d <num>] [-t <tests>] [-C <num>]
        [file.jpg ...]

┌──(root㉿kali)-[/home/kali/Rhino_CaseStudy/rhino/steg]
└─# git clone https://github.com/poizan42/stegdetect
Cloning into 'stegdetect'...
remote: Enumerating objects: 419, done.
remote: Total 419 (delta 0), reused 0 (delta 0), pack-reused 419 (from 1)
Receiving objects: 100% (419/419), 1.38 MiB | 2.78 MiB/s, done.
Resolving deltas: 100% (59/59), done.

┌──(root㉿kali)-[/home/kali/Rhino_CaseStudy/rhino/steg]
└─# ls
f0104057.jpg  f0104249.jpg  f0105065.jpg  f0105873.jpg  f0335081.jpg  stegdetect

┌──(root㉿kali)-[/home/kali/Rhino_CaseStudy/rhino/steg]
└─# cd stegdetect

┌──(root㉿kali)-[/home/…/Rhino_CaseStudy/rhino/steg/stegdetect]
└─# ls
acconfig.h  break_jphide.c    common.c      dct.c             install-sh      missing         stegcompare.c
aclocal.m4  break_jphide.h    common.h      dct.h             jpeg-6b         mkinstalldirs   stegdeimage.c
arc4.c      break_jsteg.c     compat        dirname.c         jphide_table.c  rpp.c           stegdetect.1
arc4.h      break_jsteg.h     config.guess  discrimination.c  jphide_table.h  rpp.h           stegdetect.c
bf-586.s    break_outguess.c  config.h.in   discrimination.h  jutil.c         rules.c         strlcat.c
bf_enc.c    break_outguess.h  config.sub    err.c             jutil.h         rules.h         strlcpy.c
bf_locl.h   cfg.c             configure     extraction.c      Makefile.am     rules.ini       util.c
bf_pi.h     cfg.h             configure.in  extraction.h      Makefile.in     stamp-h.in      xsteg.c
bf_skey.c   chi2cdf.c         db.c          f5.c              math.c          stegbreak.1     xsteg.h
blowfish.h  chi2cdf.h         db.h          file              md5.c           stegbreak.c     xsteg_xpm.c
```

give the image files for stegdetect in order to detect the stegnography is present

```
┌──(root㉿kali)-[/home/kali/Rhino_CaseStudy/rhino/steg]
└─# stegdetect *.jpg
f0104057.jpg : negative
f0104249.jpg : jphide(*)
f0105065.jpg : skipped (false positive likely)
f0105873.jpg : negative
f0335081.jpg : negative

┌──(root㉿kali)-[/home/kali/Rhino_CaseStudy/rhino/steg]
└─# stegdetect  -V 3 *.jpg
Stegdetect Version 0.6.1

┌──(root㉿kali)-[/home/kali/Rhino_CaseStudy/rhino/steg]
└─# stegdetect  -s3 *.jpg
f0104057.jpg : negative
f0104249.jpg : jphide(**)
f0105065.jpg : skipped (false positive likely)
f0105873.jpg : jphide(*)
f0335081.jpg : jphide(*)
```

-s3 is for increasing the severity to level 3 or depth

by this we came to know that the images are having stegnography in them.

Break the password it with stegbreak

•Abrute-force dictionary attack on embedded JPG images

•Shipped with stegdetect

•Target on embedding used by

•Outguess

•Jphide, and

•steg-shell.

•Rules on how to manipulate words for the dictionary attack

•/usr/local/share/stegbreak/rules.ini, from John the Ripper.

```
┌──(root㉿kali)-[/home/kali/Rhino_CaseStudy/rhino/steg]
└─# ls
f0104057.jpg  f0104249.jpg  f0105065.jpg  f0105873.jpg  f0335081.jpg

┌──(root㉿kali)-[/home/kali/Rhino_CaseStudy/rhino/steg]
└─# stegbreak -f rockyou.txt ./f0104249.jpg
Error in /usr/local/share/stegbreak/rules.ini at line 1
```

By default the rules.ini file is not available in the path or in the stegbreak
so we need to we need to set it by first getting the

the rules.ini is present in the stegdetect

```
┌──(root㉿kali)-[/home/kali/Rhino_CaseStudy/rhino/steg]
└─# cd stegdetect

┌──(root㉿kali)-[/home/…/Rhino_CaseStudy/rhino/steg/stegdetect]
└─# ls
acconfig.h  bf_enc.c    break_jphide.c    break_outguess.h  common.c      config.sub    dct.c            err.c         file          jphide_table.h  math.c         rpp.c      stamp-h.in     stegdetect    strlcpy.c
aclocal.m4  bf_locl.h   break_jphide.h    cfg.c             common.h      configure     dct.h            extraction.c  index.html    jutil.c         md5.c          rpp.h      stegbreak.1    stegdetect.1  util.c
arc4.c      bf_pi.h     break_jsteg.c     cfg.h             compat        configure.in  dirname.c        extraction.h  install-sh    jutil.h         missing        rules.c    stegbreak.c    stegdetect.c  xsteg.c
arc4.h      bf_skey.c   break_jsteg.h     chi2cdf.c         config.guess  db.c          discrimination.c f0104249.jpg  jpeg-6b       Makefile.am     mkinstalldirs  rules.h    stegcompare.c  stegseekexe   xsteg.h
bf-586.s    blowfish.h  break_outguess.c  chi2cdf.h         config.h.in   db.h          discrimination.h f5.c          jphide_table.c Makefile.in     rockyou.txt    rules.ini  stegdeimage.c  strlcat.c     xsteg_xpm.c

┌──(root㉿kali)-[/home/…/Rhino_CaseStudy/rhino/steg/stegdetect]
└─# ls rules.ini
rules.ini

┌──(root㉿kali)-[/home/…/Rhino_CaseStudy/rhino/steg/stegdetect]
└─#
```

```
┌──(root㉿kali)-[/]
└─# cd usr/local/share

┌──(root㉿kali)-[/usr/local/share]
└─# ls
ca-certificates  fonts  man  sgml  stegbreak  texmf  xml  zsh

┌──(root㉿kali)-[/usr/local/share]
└─# cd stegbreak

┌──(root㉿kali)-[/usr/local/share/stegbreak]
└─# ls
rules.ini

┌──(root㉿kali)-[/usr/local/share/stegbreak]
└─#
```

copy the file rules.ini to the /usr/local/share/stegbreak path so that we can run the command for cracking the passwords with any error by the system

The rockyou.txt can be downloaded by : https://github.com/brannodorsey/naive-hashcat/releases/download/data/rockyou.txt
since the rockyou.txt is in stegdetect we need to copy all the image file from steg to stegdetect directory and execute the stegbreak command with the rockyou.txt file for bruteforcing

stegbreak -f rockyou.txt * .jpg

```
┌──(root㉿kali)-[/home/kali/Rhino_CaseStudy/rhino/steg]
└─# cp f0104057.jpg stegdetect

┌──(root㉿kali)-[/home/kali/Rhino_CaseStudy/rhino/steg]
└─# cp f0105065.jpg stegdetect

┌──(root㉿kali)-[/home/kali/Rhino_CaseStudy/rhino/steg]
└─# cp f0105873.jpg stegdetect

┌──(root㉿kali)-[/home/kali/Rhino_CaseStudy/rhino/steg]
└─# cp f0335081.jpg stegdetect

┌──(root㉿kali)-[/home/kali/Rhino_CaseStudy/rhino/steg]
└─# cd stegdetect

┌──(root㉿kali)-[/home/…/Rhino_CaseStudy/rhino/steg/stegdetect]
└─# ls
acconfig.h      break_jsteg.c     config.h.in       extraction.c  jphide_table.c  rpp.h          stegseekexe
aclocal.m4      break_jsteg.h     config.sub        extraction.h  jphide_table.h  rules.c        strlcat.c
arc4.c          break_outguess.c  configure         f0104057.jpg  jutil.c         rules.h        strlcpy.c
arc4.h          break_outguess.h  configure.in      f0104249.jpg  jutil.h         rules.ini      util.c
bf-586.s        cfg.c             db.c              f0105065.jpg  Makefile.am     stamp-h.in     xsteg.c
bf_enc.c        cfg.h             db.h              f0105873.jpg  Makefile.in     stegbreak.1    xsteg.h
bf_locl.h       chi2cdf.c         dct.c             f0335081.jpg  math.c          stegbreak.c    xsteg_xpm.c
bf_pi.h         chi2cdf.h         dct.h             f5.c          md5.c           stegcompare.c
bf_skey.c       common.c          dirname.c         file          missing         stegdeimage.c
blowfish.h      common.h          discrimination.c  index.html    mkinstalldirs   stegdetect
break_jphide.c  compat            discrimination.h  install-sh    rockyou.txt     stegdetect.1
break_jphide.h  config.guess      err.c             jpeg-6b       rpp.c           stegdetect.c

┌──(root㉿kali)-[/home/…/Rhino_CaseStudy/rhino/steg/stegdetect]
└─# stegbreak -f rockyou.txt *.jpg
Loaded 5 files...
f0105065.jpg : jphide[v5](gator)
f0104249.jpg : jphide[v5](gumbo)
```

Using Steghide/stegseek to recover the photos

for this create stegseek directory and then download the following
download and unzip jphide
wget -q ftp://gwdg.de/pub/linux/misc/ppdd/jphs_05.zip

```
┌──(root㉿kali)-[/home/…/rhino/steg/stegdetect/stegseekexe]
└─# wget -q ftp://ftp.gwdg.de/pub/linux/misc/ppdd/jphs_05.zip

┌──(root㉿kali)-[/home/…/rhino/steg/stegdetect/stegseekexe]
└─# ls
f0104249.jpg  jphs_05.zip

┌──(root㉿kali)-[/home/…/rhino/steg/stegdetect/stegseekexe]
└─# unzip jphs_05.zip
Archive:  jphs_05.zip
 extracting: jphs05.zip
 extracting: jphs05.zip.sig

┌──(root㉿kali)-[/home/…/rhino/steg/stegdetect/stegseekexe]
└─# unzip jphs05.zip
Archive:  jphs05.zip
  inflating: jphide.exe
  inflating: jpseek.exe
  inflating: Jphswin.exe
  inflating: Readme.txt

┌──(root㉿kali)-[/home/…/rhino/steg/stegdetect/stegseekexe]
└─#
```

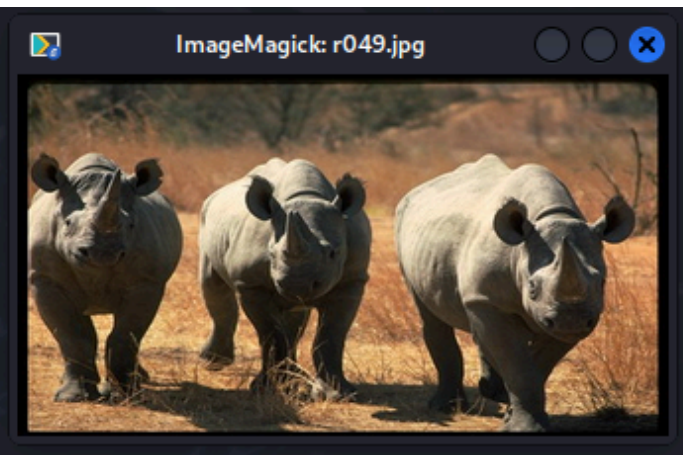
Using this jpseek.exe we can recover the image behind the front one by giving the password we had got by stegbreak
JPHIDE.EXE is a DOS program to hide a data file in a jpeg file.
JPSEEK.EXE is a DOS program to recover a file hidden with JPHIDE.EXE

Upon executing the jpseek.exe for the image file and password we can get the output image file as r049.jpg

```
┌──(root㉿kali)-[/home/…/rhino/steg/stegdetect/stegseekexe]
└─# wine jpseek.exe f0104249.jpg r049.jpg

Welcome to jpseek Rev 0.51
 (c) 1998 Allan Latham <alatham@flexsys-group.com>
This program is freeware.
No charge is made for its use.
Use at your own risk. No liability accepted whatever happens.
Contains cryptogaphy which may be subject to local laws.

Passphrase:

┌──(root㉿kali)-[/home/…/rhino/steg/stegdetect/stegseekexe]
└─# ls
f0104249.jpg  jphide.exe  jphs_05.zip  jphs05.zip  jphs05.zip.sig  Jphswin.exe  jpseek.exe  r049.jpg  Readme.txt
```

Display r049.jpg

```
┌──(root㉿kali)-[/home/…/rhino/steg/stegdetect/stegseekexe]
└─# wine jpseek.exe f0104249.jpg r049.jpg

Welcome to jpseek Rev 0.51
 (c) 1998 Allan Latham <alatham@flexsys-group.com>
This program is freeware.
No charge is made for its use.
Use at your own risk. No liability accepted whatever happens.
Contains cryptogaphy which may be subject to local laws.

Passphrase:

┌──(root㉿kali)-[/home/…/rhino/steg/stegdetect/stegseekexe]
└─# ls
f0104249.jpg  jphide.exe  jphs_05.zip  jphs05.zip  jphs05.zip.sig  Jphswin.exe  jpseek.exe  r049.jpg  Readme.txt

┌──(root㉿kali)-[/home/…/rhino/steg/stegdetect/stegseekexe]
└─# display r049.jpg
```




```
┌──(root㉿kali)-[/home/…/rhino/steg/stegdetect/stegseekexe]
└─# wine jpseek.exe f0105065.jpg r065.jpg

Welcome to jpseek Rev 0.51
 (c) 1998 Allan Latham <alatham@flexsys-group.com>
This program is freeware.
No charge is made for its use.
Use at your own risk. No liability accepted whatever happens.
Contains cryptogaphy which may be subject to local laws.

Passphrase:

┌──(root㉿kali)-[/home/…/rhino/steg/stegdetect/stegseekexe]
└─# ls
f0104249.jpg  jphide.exe   jphs05.zip      Jphswin.exe  r049.jpg  Readme.txt
f0105065.jpg  jphs_05.zip  jphs05.zip.sig  jpseek.exe   r065.jpg

┌──(root㉿kali)-[/home/…/rhino/steg/stegdetect/stegseekexe]
└─# display r065.jpg
```

Examination Phase 2 : Tracing_Logs FTP Traffic

Here We Will see into the Log files with wireshark for further analysis

Look into Tools and check for Credentials For analysis of the sessions

We can see the Session in the credentials of rhino.log where there are three different sessions examining the sessions

kali-linux-2024.2-virtualbox-amd64 [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

rhino.log

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
1527	179.012224	137.30.120.40	137.30.122.253	TCP	62	21 → 1655 [SYN, ACK] Seq=0 Ack=1 Win=49640
1528	179.012284	137.30.122.253	137.30.120.40	TCP	54	1655 → 21 [ACK] Seq=1 Ack=1 Win=64240 Len=0
1529	179.040214	137.30.120.40	137.30.122.253	FTP	82	Response: 220 cook FTP server ready.
1530	179.189473	137.30.122.253	137.30.120.40	TCP	54	1655 → 21 [ACK] Seq=1 Ack=29 Win=64212 Len=
1531	182.548193	128.138.182.127	137.30.123.164	TCP	62	4675 → 2745 [SYN] Seq=0 Win=16384 Len=0 MSS
1532	182.640647	137.30.122.253	137.30.120.40	FTP	66	Request: USER gnome
1533	182.640847	137.30.120.40	137.30.122.253	TCP	60	21 → 1655 [ACK] Seq=29 Ack=13 Win=49640 Len
1534	182.644970	137.30.120.40	137.30.122.253	FTP	88	Response: 331 Password required for gnome.
1535	182.800991	137.30.122.253	137.30.120.40	TCP	54	1655 → 21 [ACK] Seq=13 Ack=63 Win=64178 Len
1536	184.667754	137.30.122.253	137.30.120.40	FTP	69	Request: PASS gnome123
1537	184.667952	137.30.120.40	137.30.122.253	TCP	60	21 → 1655 [ACK] Seq=63 Ack=28 Win=49640 Len
1538	184.748946	137.30.120.40	137.30.122.253	FTP	81	Response: 230 User gnome logged in.
1539	184.907708	137.30.122.253	137.30.120.40	TCP	54	1655 → 21 [ACK] Seq=28 Ack=90 Win=64151 Len
1540	185.602553	137.30.122.253	137.30.120.40	FTP	62	Request: TYPE I
1541	185.602818	137.30.120.40	137.30.122.253	FTP	74	Response: 200 Type set to I.
1542	185.710258	137.30.122.253	137.30.120.40	TCP	54	1655 → 21 [ACK] Seq=36 Ack=110 Win=64131 Le
1543	185.730515	128.138.182.127	137.30			
1544	188.994914	137.30.122.253	137.30			
1545	188.995519	137.30.120.40	137.30			
1546	188.996081	137.30.122.253	137.30			

```
Frame 1536: 69 bytes on wire (552 bits), 69 byte
Ethernet II, Src: Apple_cc:57:92 (00:03:93:cc:57
Internet Protocol Version 4, Src: 137.30.122.253
Transmission Control Protocol, Src Port: 1655, D
File Transfer Protocol (FTP)
  PASS gnome123\r\n
     Request command: PASS
     Request arg: gnome123
  [Current working directory: ]
```

Wireshark · Credentials · rhino.log

Packet N	Protocol	Username	Additional Info
1536	FTP	gnome	Username in packet: 1532
1629	FTP	gnome	Username in packet: 1625
5637	FTP	gnome	Username in packet: 5633

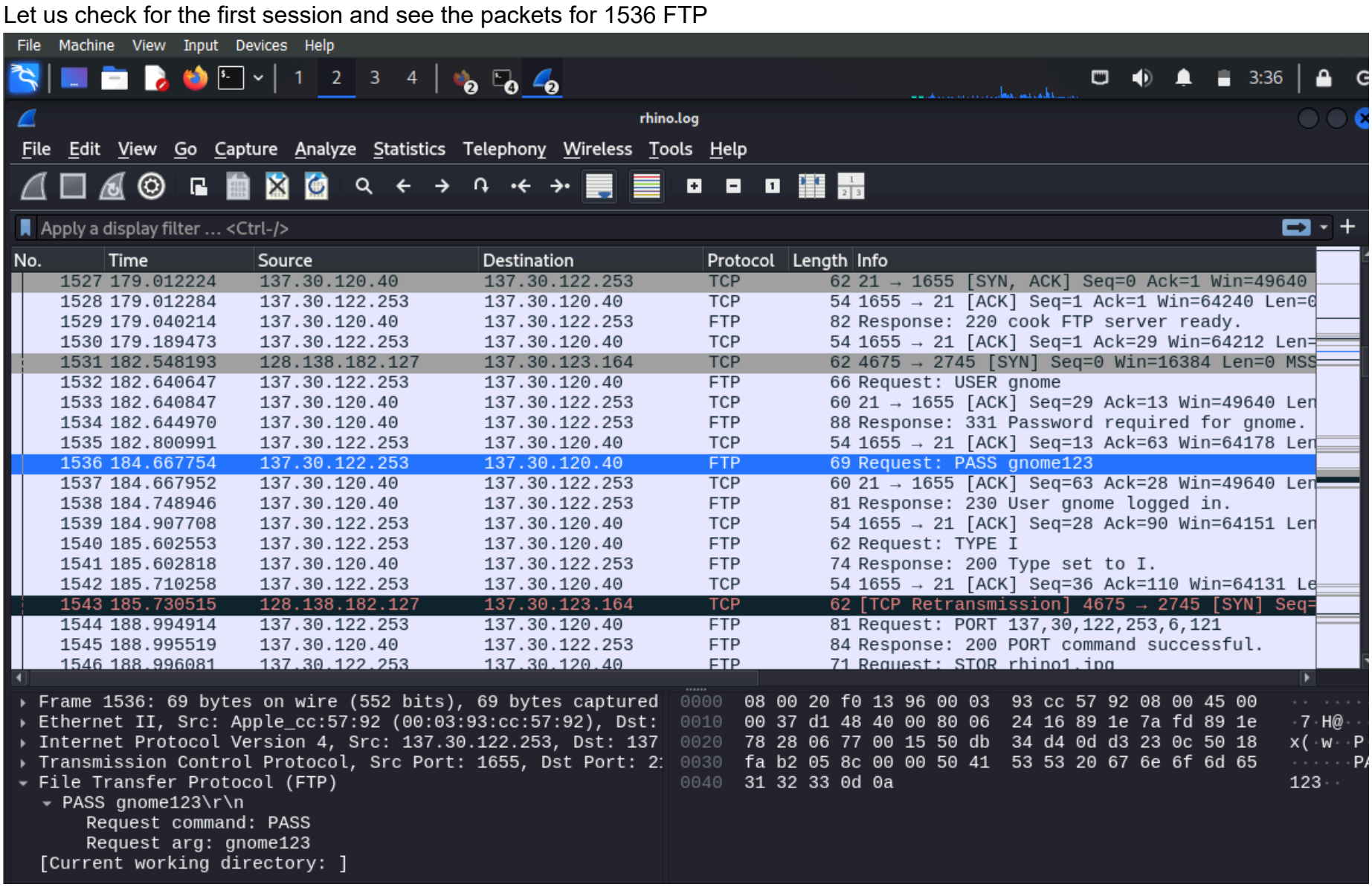
× Close

rhino.log

Default

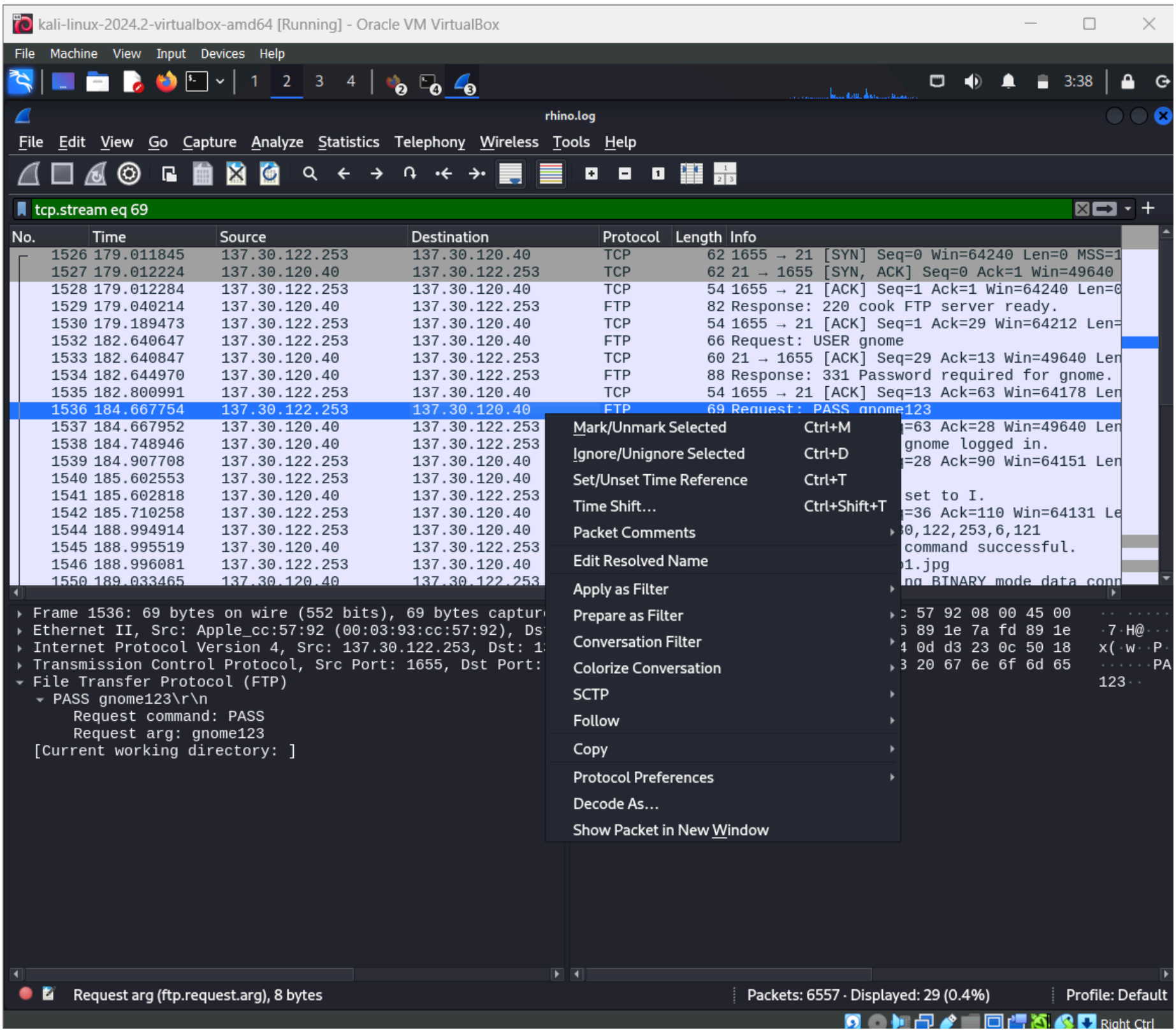
Right Ctrl

Let us check for the first session and see the packets for 1536 FTP

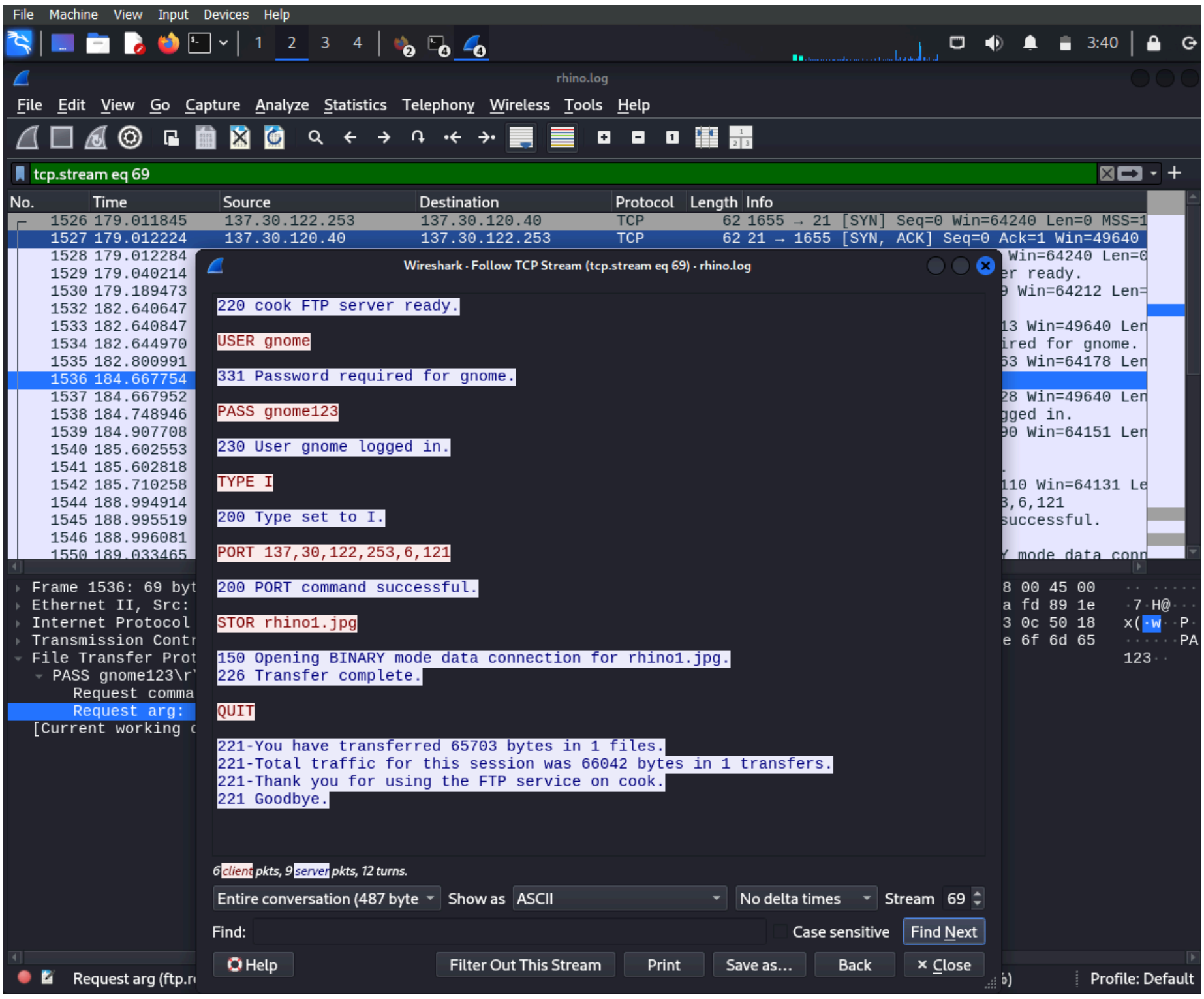


By seeing this we got to know that the suspect gave the credentials and passed the password

by further examining the packet by following it's Tcp stream try to gather any useful information



We can see the converstation between the suspect and the server by seeing the ASCII format
in that try to see on the packet of opening BINARY mode data slect it the woreshark open s the packet and examine it .



By that packet seen below we can check that the Response has been done and the next step is to set a filter "Time Display Format" and keep it in UTC Time and date format so that we can see the next conversation by which we can get the FTP DATA.

kali-linux-2024.2-virtualbox-amd64 [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

1 2 3 4 3:44

rhino.log

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

tcp.stream eq 69

No.	Time	Source	Destination	Protocol	Length	Info
1537	184.667952	137.30.120.40	137.30.122.253	TCP	60	21 → 1655 [ACK] Seq=63 Ack=28 Win=49640 Len
1538	184.748946	137.30.120.40	137.30.122.253	FTP	81	Response: 230 User gnome logged in.
1539	184.907708	137.30.122.253	137.30.120.40	TCP	54	1655 → 21 [ACK] Seq=28 Ack=90 Win=64151 Len
1540	185.602553	137.30.122.253	137.30.120.40	FTP	62	Request: TYPE I
1541	185.602818	137.30.120.40	137.30.122.253	FTP	74	Response: 200 Type set to I.
1542	185.710258	137.30.122.253	137.30.120.40	TCP	54	1655 → 21 [ACK] Seq=36 Ack=110 Win=64131 Le
1544	188.994914	137.30.122.253	137.30.120.40	FTP	81	Request: PORT 137,30,122,253,6,121
1545	188.995519	137.30.120.40	137.30.122.253	FTP	84	Response: 200 PORT command successful.
1546	188.996081	137.30.122.253	137.30.120.40	FTP	71	Request: STOR rhino1.jpg
1550	189.033465	137.30.120.40	137.30.122.253	FTP	111	Response: 150 Opening BINARY mode data conn
1611	189.221464	137.30.122.253	137.30.120.40	TCP	54	1655 → 21 [ACK] Seq=80 Ack=197 Win=64044 Le
1612	189.221711	137.30.120.40	137.30.122.253	FTP	78	Response: 226 Transfer complete.
1613	189.422115	137.30.122.253	137.30.120.40	TCP	54	1655 → 21 [ACK] Seq=80 Ack=221 Win=64020 Le
1614	194.426879	137.30.122.253	137.30.120.40	FTP	60	Request: QUIT
1615	194.427484	137.30.120.40	137.30.122.253	FTP	104	Response: 221-You have transferred 65703 by
1616	194.432107	137.30.120.40	137.30.122.253	FTP	186	Response: 221-Total traffic for this sessio
1617	194.432192	137.30.122.253	137.30.120.40	TCP	54	1655 → 21 [ACK] Seq=86 Ack=404 Win=63838 Le
1618	194.433622	137.30.122.253	137.30.120.40	TCP	54	1655 → 21 [FIN, ACK] Seq=86 Ack=404 Win=638
1619	194.433807	137.30.120.40	137.30.122.253	TCP	60	21 → 1655 [ACK] Seq=404 Ack=87 Win=49640 Le

```
▸ Frame 1550: 111 bytes on wire (888 bits), 111 bytes captured (888 bits)
▸ Ethernet II, Src: Oracle_f0:13:96 (08:00:20:f0:13:96), Dst: Apple_cc:57:92 (00:03:93:cc:57:92)
▸ Internet Protocol Version 4, Src: 137.30.120.40, Dst: 137.30.122.253
▸ Transmission Control Protocol, Src Port: 21, Dst Port: 1655, Seq: 140, Ack: 80, Len: 57
▾ File Transfer Protocol (FTP)
  ▾ 150 Opening BINARY mode data connection for rhino1.jpg.\r\n
      Response code: File status okay; about to open data connection (150)
      Response arg: Opening BINARY mode data connection for rhino1.jpg.
  [Current working directory: ]
```

```
0000  00 03 93 cc 57 92 08
0010  00 61 7f f4 40 00 3c
0020  7a fd 00 15 06 77 0d
0030  c1 e8 5f 90 00 00 31
0040  67 20 42 49 4e 41 52
0050  74 61 20 63 6f 6e 6e
0060  72 20 72 68 69 6e 6f
```

Text item (text), 57 bytes | Packets: 6557 · Displayed: 29 (0.4%) | Profile: Default

Right Ctrl

kali-linux-2024.2-virtualbox-amd64 [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

1 2 3 4 3:45

rhino.log

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

tcp.strea

No. | Protocol | Length | Info

1537 TCP 60 21 → 1655 [ACK] Seq=63 Ack=28 Win=49640 Len
1538 FTP 81 Response: 230 User gnome logged in.
1539 TCP 54 1655 → 21 [ACK] Seq=28 Ack=90 Win=64151 Len
1540 FTP 62 Request: TYPE I
1541 FTP 74 Response: 200 Type set to I.
1542 TCP 54 1655 → 21 [ACK] Seq=36 Ack=110 Win=64131 Le
1544 FTP 81 Request: PORT 137,30,122,253,6,121
1545 FTP 84 Response: 200 PORT command successful.
1546 FTP 71 Request: STOR rhino1.jpg
1550
1611
1612
1613
1614
1615
1616
1617
1618
1619

Frame 1
Etherne
Interne
Transmi
File Tr
150 C
Res
Res
[Curren

92 08
00 3c
77 0d
00 31
41 52
6e 6e
6e 6f

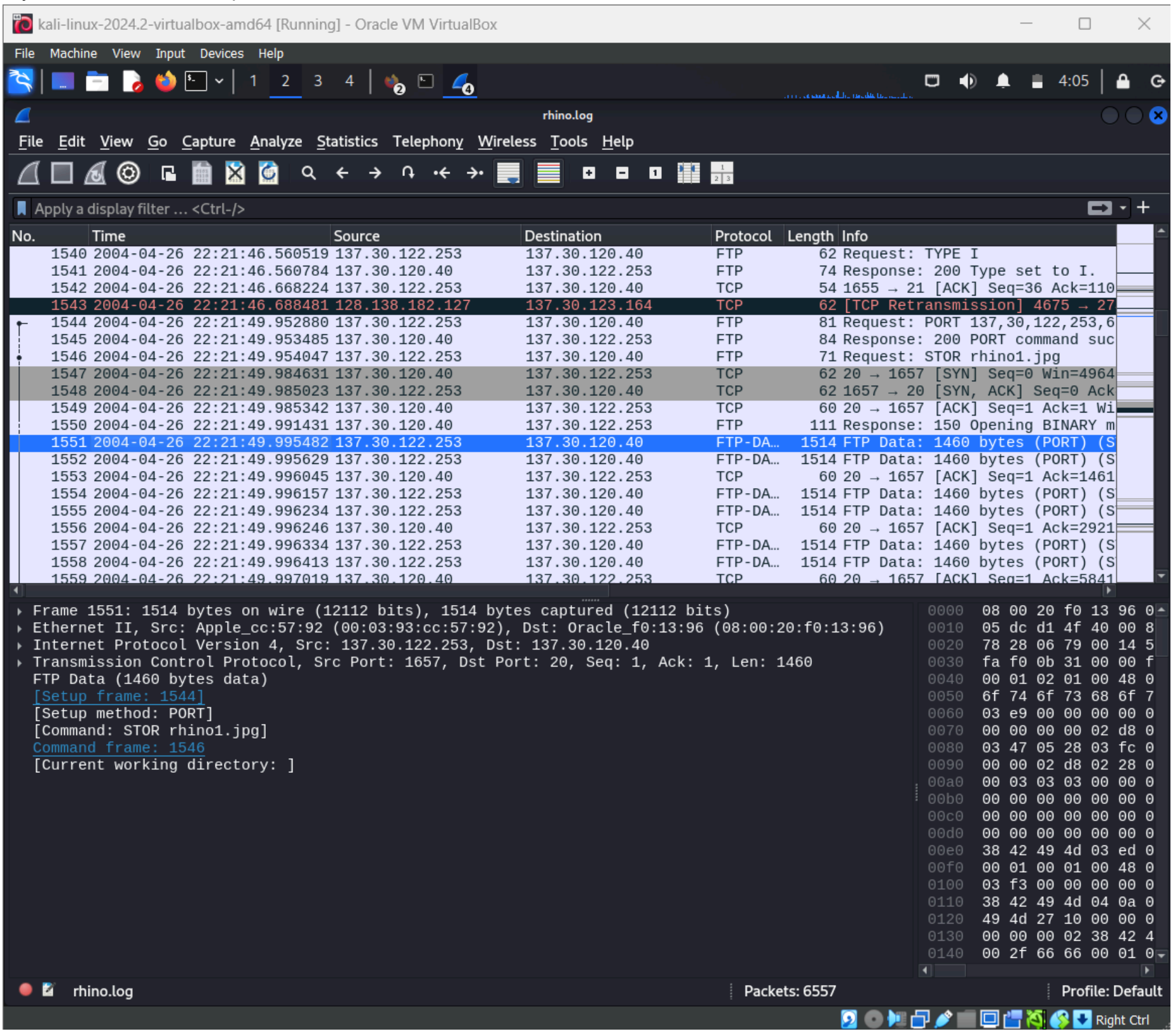
View menu:
- ✓ Main Toolbar
- ✓ Filter Toolbar
- Wireless Toolbar
- ✓ Status Bar
- Full Screen F11
- ✓ Packet List
- ✓ Packet Details
- ✓ Packet Bytes
- Packet Diagram
- Time Display Format
- Name Resolution
- Zoom
- Expand Subtrees Shift+Right
- Collapse Subtrees Shift+Left
- Expand All Ctrl+Right
- Collapse All Ctrl+Left
- Colorize Packet List
- Coloring Rules...
- Colorize Conversation
- Reset Layout Ctrl+Shift+W
- Resize Columns Ctrl+Shift+R
- Internals
- Show Packet in New Window
- Reload as File Format/Capture Ctrl+Shift+F
- Reload Ctrl+R

Time Display Format:
- Date and Time of Day (1970-01-01 01:02:03.123456) Ctrl+Alt+1
- Year, Day of Year, and Time of Day (1970/001 01:02:03.123456)
- Time of Day (01:02:03.123456) Ctrl+Alt+2
- Seconds Since 1970-01-01 Ctrl+Alt+3
- • Seconds Since First Captured Packet Ctrl+Alt+4
- Seconds Since Previous Captured Packet Ctrl+Alt+5
- Seconds Since Previous Displayed Packet Ctrl+Alt+6
- UTC Date and Time of Day (1970-01-01 01:02:03.123456) Ctrl+Alt+7
- UTC Year, Day of Year, and Time of Day (1970/001 01:02:03.123456)
- UTC Time of Day (01:02:03.123456) Ctrl+Alt+8
- • Automatic (from capture file)
- Seconds
- Tenths of a second
- Hundredths of a second
- Milliseconds
- Tenths of a millisecond
- Hundredths of a millisecond
- Microseconds
- Tenths of a microsecond
- Hundredths of a microsecond
- Nanoseconds
- Display Seconds With Hours and Minutes

Text item (text), 57 bytes | Packets: 6557 · Displayed: 29 (0.4%) | Profile: Default

Right Ctrl

Try to see the FTP DATA packet to see the file data which we wanted it is hex format



Verify the MAGIC NUMBER if it is starting with the ffd8 and ending with ffd9 is the image file and save the hex format with .jpg in this case it is rhino1.jpg and try to display it by below and we got the image .

kali-linux-2024.2-virtualbox-amd64 [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

rhino.log

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

tcp.stream eq 71

No.	Time	Source	Destination	Protocol	Length	Info
1547	2004-04-26 22:21:49.984631	137.30.120.40	137.30.122.253	TCP	62	20 → 1657 [SYN] Seq=0 Win=4964
1548	2004-04-26 22:21:49.985023	137.30.122.253	137.30.120.40	TCP	62	1657 → 20 [SYN, ACK] Seq=0 Ack
1549	2004-04-26					ck=1 Wi
1551	2004-04-26					ORT) (S
1552	2004-04-26					
1553	2004-04-26					
1554	2004-04-26					
1555	2004-04-26					
1556	2004-04-26					
1557	2004-04-26					
1558	2004-04-26					
1559	2004-04-26					
1560	2004-04-26					
1561	2004-04-26					
1562	2004-04-26					
1563	2004-04-26					
1564	2004-04-26					
1565	2004-04-26					
1566	2004-04-26					
1567	2004-04-26					

Frame 1551: 1514 b
Ethernet II, Src:
Internet Protocol
Transmission Contr
FTP Data (1460 byt
[Setup frame: 154
[Setup method: POR
[Command: STOR rhi
Command frame: 154
[Current working d

rhino.log

Wireshark · Follow TCP Stream (tcp.stream eq 71) · rhino.log

```
ffd8ffe000104a4649460001020100480048000
0000000007800030000004800480000000020
0000000002d80228000100000064000000100
6008001901900000000000000000000000000
0000000001000480000001000100480000000
494d040a0000000000010000 3842494d271000
0048002f66660001006c666600060000000000
005a0000006000000000001003500000001000
ffffffffffffffffffffffffffffffffffffff
fffffffffff03e800000000ffffffffffffff
ffffffffffffffffffffffffffffffffffffff01
0240000000003842494d04090000000013a300
0001ffd8ffe000104a4649460001020100480048
6f626520506 86f746f73686f70a820342e3000
09080c09090c110b0a0b11150f0c0c0f151815
0c0c0c0c0c0c0c0c0c0c0c0c0c0c0c0c010d0b
11110c0c0c0c0c0c110c0c0c0c0c0c0c0c0c0c
0080030122000211010311 01ffdd00040008f
05060708090a0b0100010501010101010100000
020507060805030c330100021103042112310
4307259253f0e1f163733516a2b2832644935
95c4d4e4f4a5b5c5d5e5f55666768696a6b6c
0405060707060535010002110321311204415
5315637334f1250616a2b283072635c2d2449
d4e4f4a5b5c5d5e5f55666768696a6b6c6d6e
00c11d7afd3681f1239562aebd69001027c610
7fbb63fdd0c7f7c3dff066ceb6f0408903925
d4741698899d025f76c7fba11f7beb69c75ab
c7a2d8658c3ea469a69c25ec63fdd0afbdff0
dc5ad1c8d1533834ef0db4ec0741247297b18
ee30044b208e1428e90db6971b08f506844a5
96f03d22c01ce04b4caaa7a3c30b9ffa371d083c23f77c7fba13f7afeb2cff00ac2f2e33007820bfaf5bc03f
```

0 client pkts, 46 server pkts, 0 turns.

Entire conversation (65 kB) | Show as Raw | No delta times | Stream 71

Find: | Case sensitive | Find Next

Help | Filter Out This Stream | Print | Save as... | Back | × Close

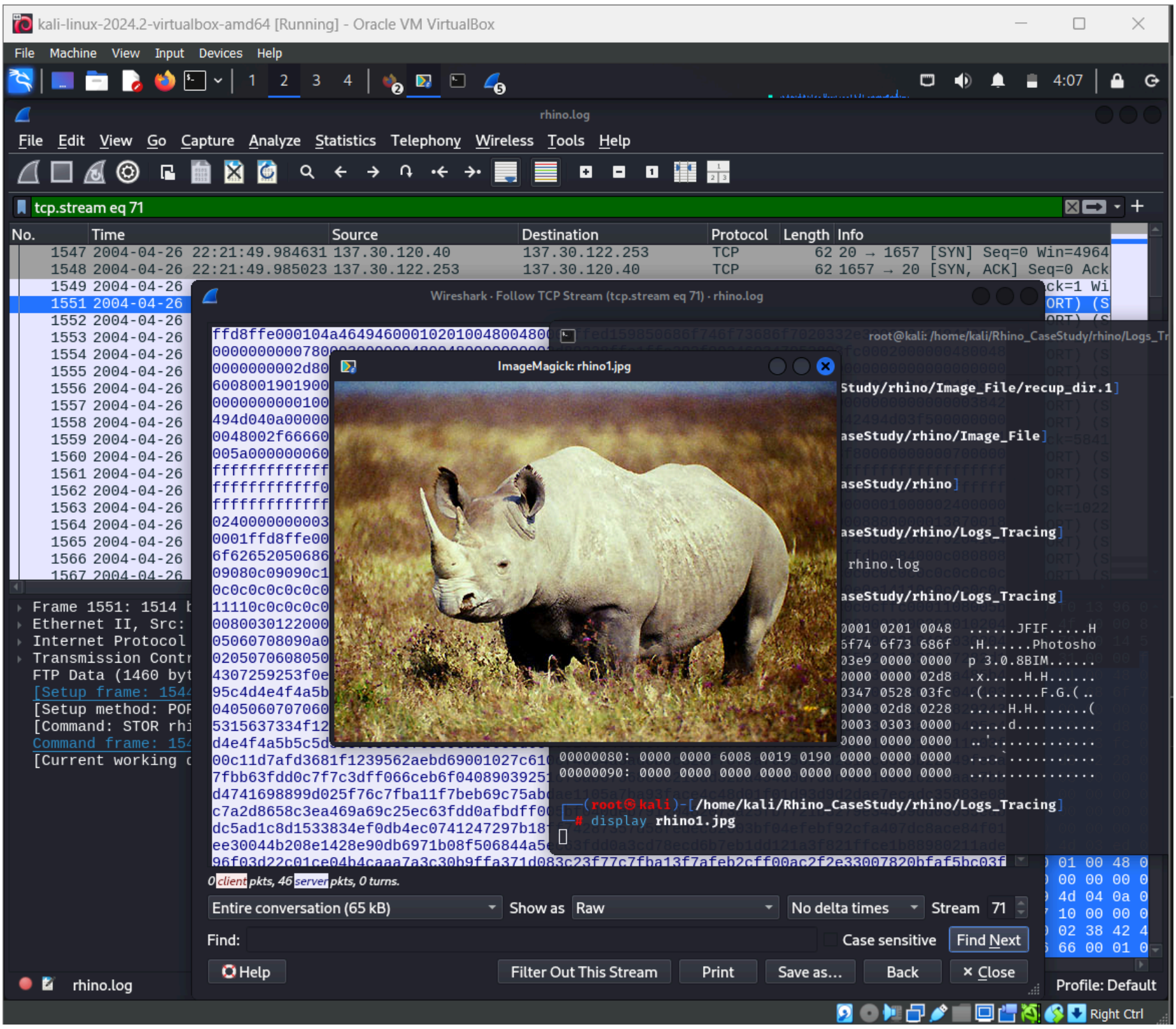
root@kali: /home/kali/Rhino_CaseStudy/rhino/Logs_T

File Actions Edit View Help

```
┌──(root㉿kali)-[/home/…/Rhino_CaseStudy/rhino/Image_File/recup_dir.1]
└─# cd ..

┌──(root㉿kali)-[/home/kali/Rhino_CaseStudy/rhino/Image_File]
└─# cd ..

┌──(root㉿kali)-[/home/kali/Rhino_CaseStudy/rhino]
└─# cd Logs_Tracing

┌──(root㉿kali)-[/home/kali/Rhino_CaseStudy/rhino/Logs_Tracing]
└─# ls
rhino1.jpg  rhino2.log  rhino3.log  rhino.log

┌──(root㉿kali)-[/home/kali/Rhino_CaseStudy/rhino/Logs_Tracing]
└─# xxd rhino1.jpg | head
00000000: ffd8 ffe0 0010 4a46 4946 0001 0201 0048  ......JFIF.....H
00000010: 0048 0000 ffed 1598 5068 6f74 6f73 686f  .H......Photosho
00000020: 7020 332e 3000 3842 494d 03e9 0000 0000  p 3.0.8BIM......
00000030: 0078 0003 0000 0048 0048 0000 0000 02d8  .x.....H.H......
00000040: 0228 ffe1 ffe2 02f9 0246 0347 0528 03fc  .(.......F.G.(..
00000050: 0002 0000 0048 0048 0000 0000 02d8 0228  .....H.H.......(
00000060: 0001 0000 0064 0000 0001 0003 0303 0000  .....d..........
00000070: 0001 270f 0001 0001 0000 0000 0000 0000  ..'.............
00000080: 0000 0000 6008 0019 0190 0000 0000 0000  ....`...........
00000090: 0000 0000 0000 0000 0000 0000 0000 0000  ................

┌──(root㉿kali)-[/home/kali/Rhino_CaseStudy/rhino/Logs_Tracing]
└─#
```

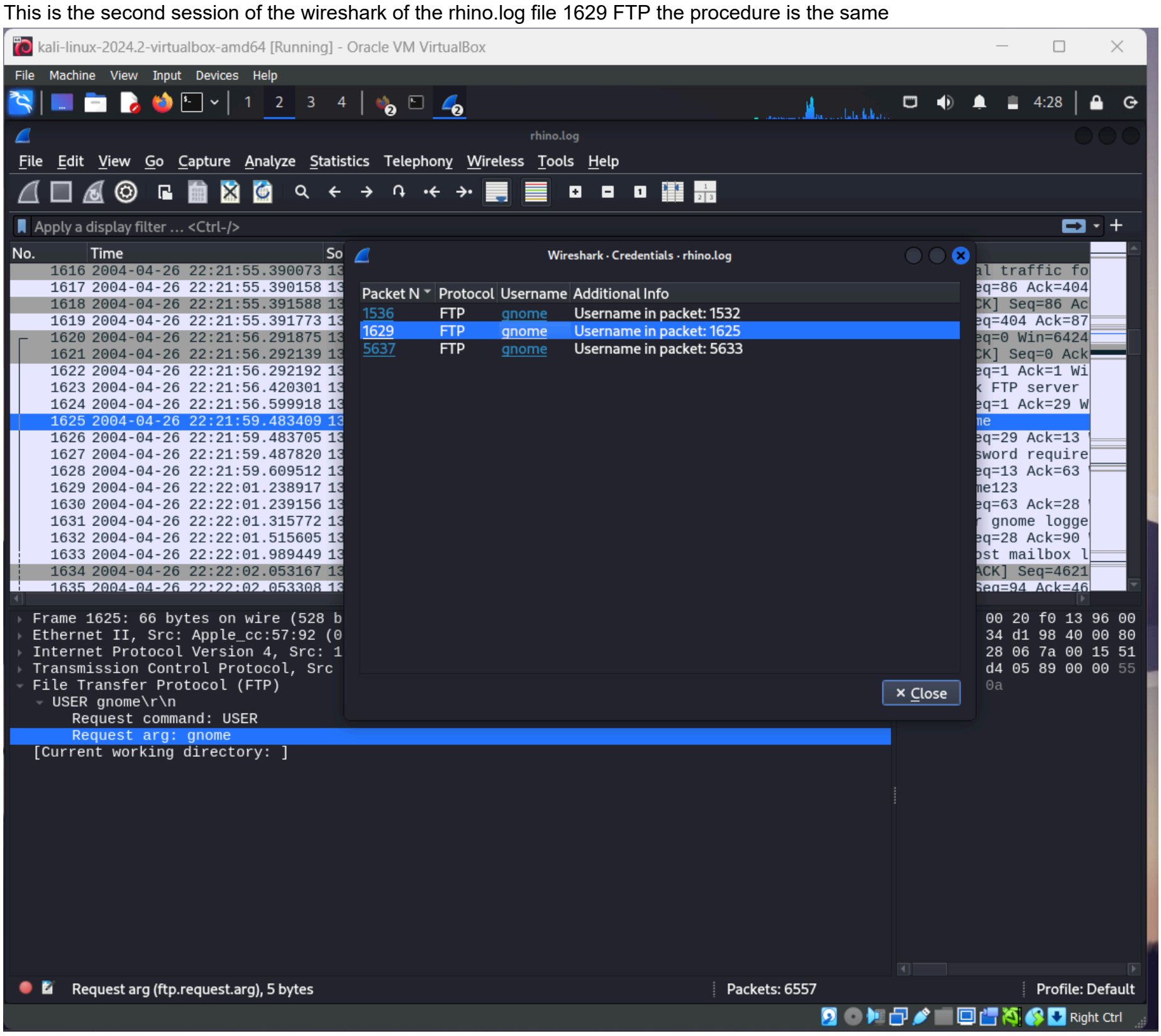
Profile: Default

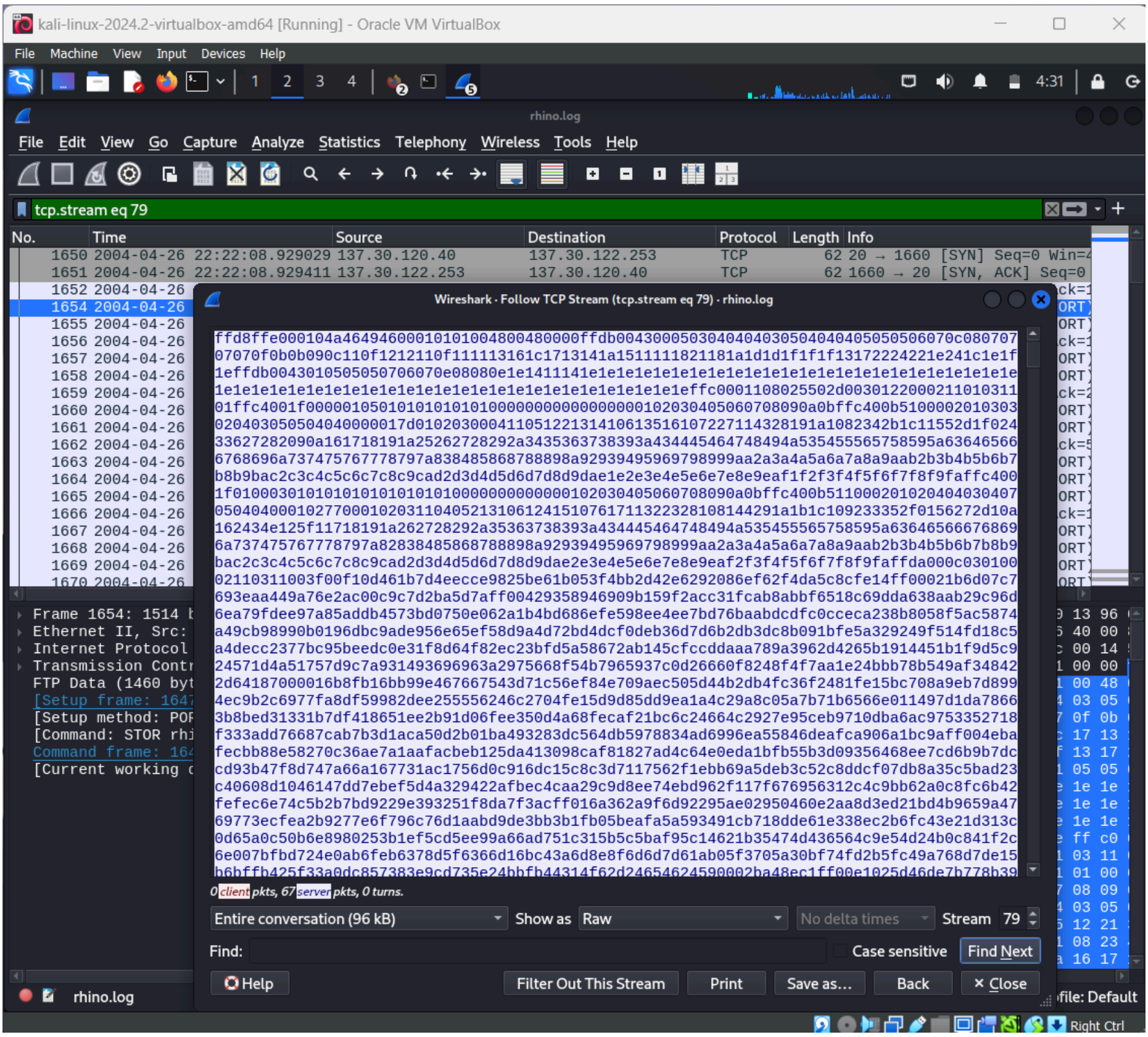
Right Ctrl


kali-linux-2024.2-virtualbox-amd64 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
rhino.log
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
tcp.stream eq 71
No. Time Source Destination Protocol Length Info
1547 2004-04-26 22:21:49.984631 137.30.120.40 137.30.122.253 TCP 62 20 → 1657 [SYN] Seq=0 Win=4964
1548 2004-04-26 22:21:49.985023 137.30.122.253 137.30.120.40 TCP 62 1657 → 20 [SYN, ACK] Seq=0 Ack
Wireshark · Follow TCP Stream (tcp.stream eq 71) · rhino.log
ImageMagick: rhino1.jpg
root@kali: /home/kali/Rhino_CaseStudy/rhino/Logs_Tr
(root㉿kali)-[/home/kali/Rhino_CaseStudy/rhino/Logs_Tracing]
display rhino1.jpg
Entire conversation (65 kB)
Show as Raw
No delta times
Stream 71
Find:
Case sensitive
Find Next
Help
Filter Out This Stream
Print
Save as...
Back
Close
Profile: Default
Right Ctrl

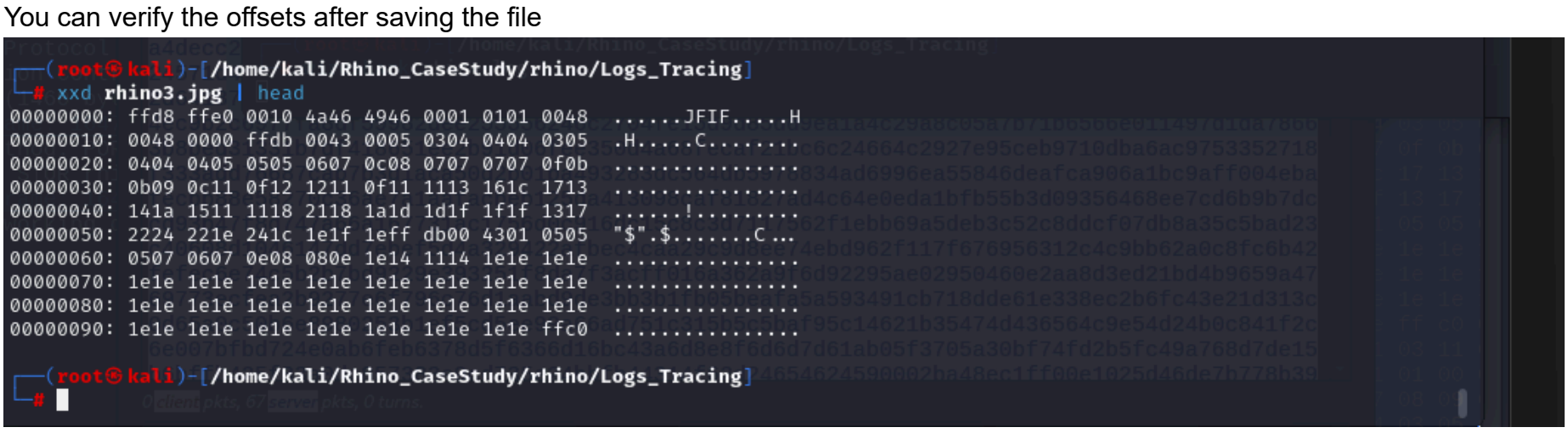
2nd session

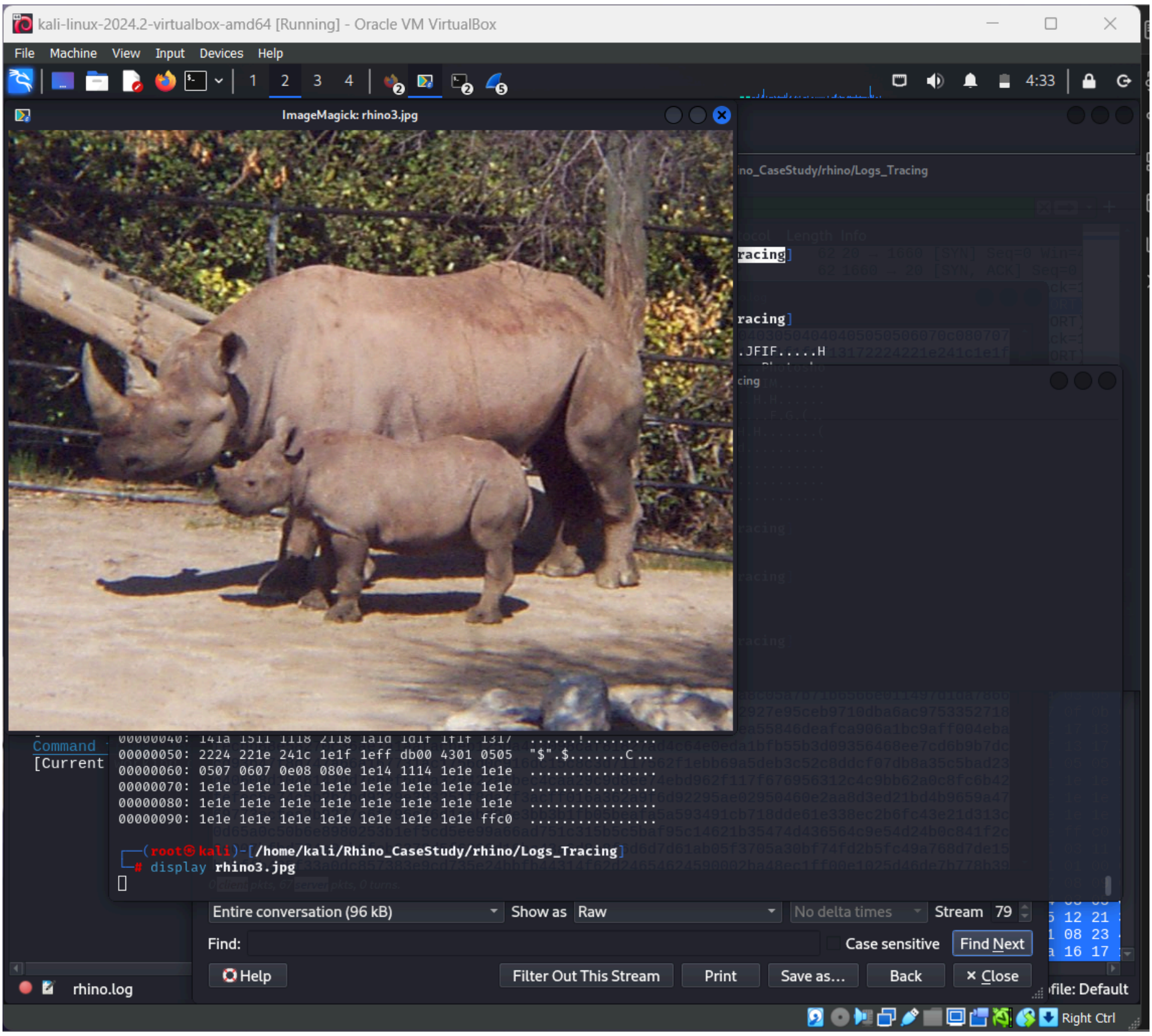
This is the second session of the wireshark of the rhino.log file 1629 FTP the procedure is the same





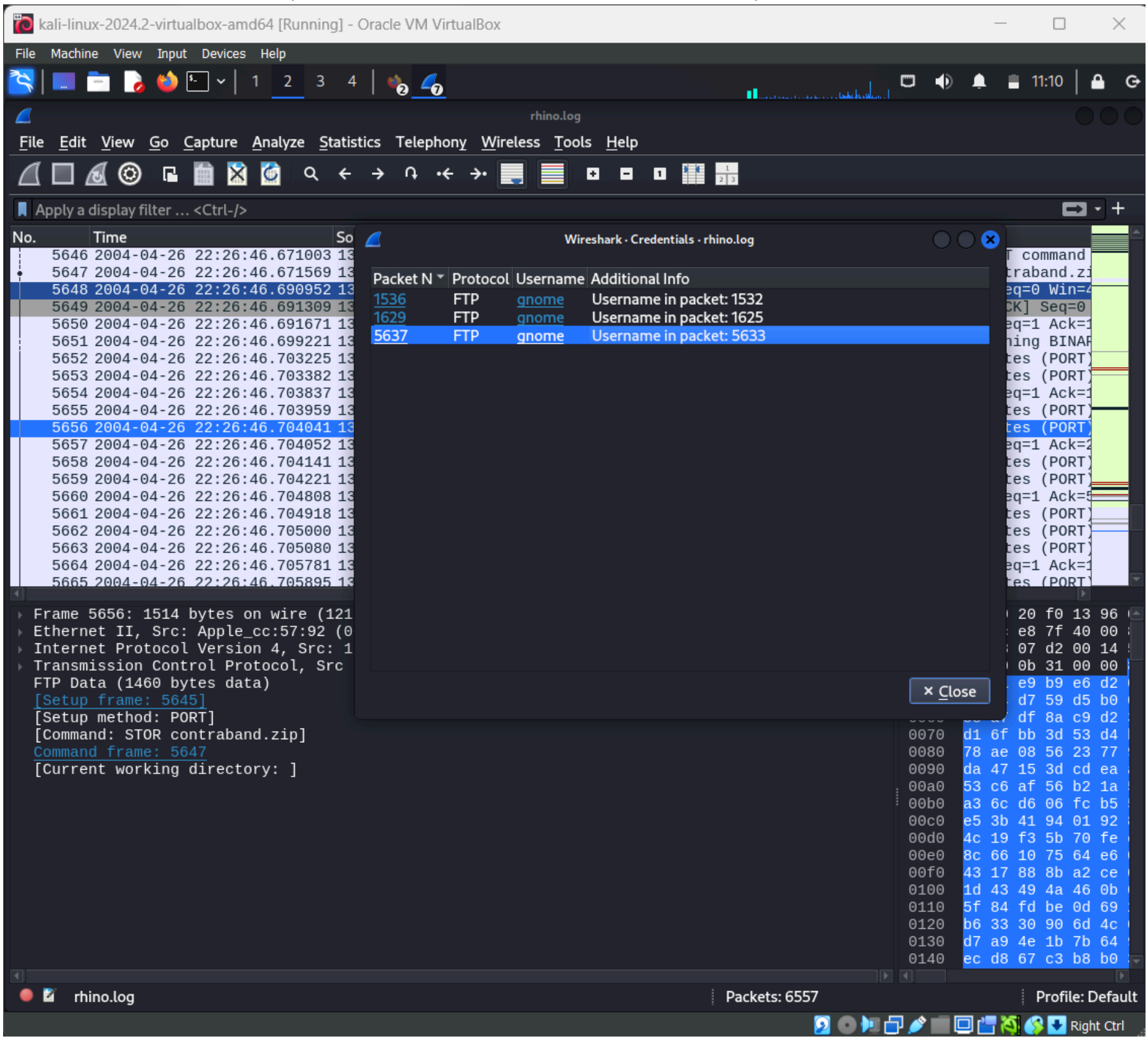
You can verify the offsets after saving the file





By second session we got other image file

3rd session 5637 FTP the same procedure but here the content is in file format in a zip file



kali-linux-2024.2-virtualbox-amd64 [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

1 2 3 4 11:11

rhino.log

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
5655	2004-04-26 22:26:46.703959	137.30.122.253	137.30.120.40	FTP-DA...	1514	FTP Data: 1460 bytes (PORT)
5656	2004-04-26 22:26:46.704041	137.30.122.253	[illegible]	[illegible]	[illegible]	Data: 1460 bytes (PORT)
5657	2004-04-26 22:26:46.704052	137.30.120.40				→ 2002 [ACK] Seq=1 Ack=2
5658	2004-04-26 22:26:46.704141	137.30.122.253				Data: 1460 bytes (PORT)
5659	2004-04-26 22:26:46.704221	137.30.122.253				Data: 1460 bytes (PORT)
5660	2004-04-26 22:26:46.704808	137.30.120.40				→ 2002 [ACK] Seq=1 Ack=5
5661	2004-04-26 22:26:46.704918	137.30.122.253				Data: 1460 bytes (PORT)
5662	2004-04-26 22:26:46.705000	137.30.122.253				Data: 1460 bytes (PORT)
5663	2004-04-26 22:26:46.705080	137.30.122.253				Data: 1460 bytes (PORT)
5664	2004-04-26 22:26:46.705781	137.30.120.40				→ 2002 [ACK] Seq=1 Ack=1
5665	2004-04-26 22:26:46.705895	137.30.122.253				Data: 1460 bytes (PORT)
5666	2004-04-26 22:26:46.705980	137.30.122.253				Data: 1460 bytes (PORT)
5667	2004-04-26 22:26:46.706060	137.30.122.253				Data: 1460 bytes (PORT)
5668	2004-04-26 22:26:46.706141	137.30.122.253				Data: 1460 bytes (PORT)
5669	2004-04-26 22:26:46.719657	137.30.120.40				→ 2002 [ACK] Seq=1 Ack=1
5670	2004-04-26 22:26:46.719806	137.30.122.253				Data: 1460 bytes (PORT)
5671	2004-04-26 22:26:46.719892	137.30.122.253				Data: 1460 bytes (PORT)
5672	2004-04-26 22:26:46.719971	137.30.122.253				Data: 1460 bytes (PORT)
5673	2004-04-26 22:26:46.720051	137.30.122.253				Data: 1460 bytes (PORT)
5674	2004-04-26 22:26:46.720133	137.30.122.253				Data: 1460 bytes (PORT)

- Frame 5656: 1514 bytes on wire (12112 bits), 1514 byt
- Ethernet II, Src: Apple_cc:57:92 (00:03:93:cc:57:92),
- Internet Protocol Version 4, Src: 137.30.122.253, Dst
- Transmission Control Protocol, Src Port: 2002, Dst Po

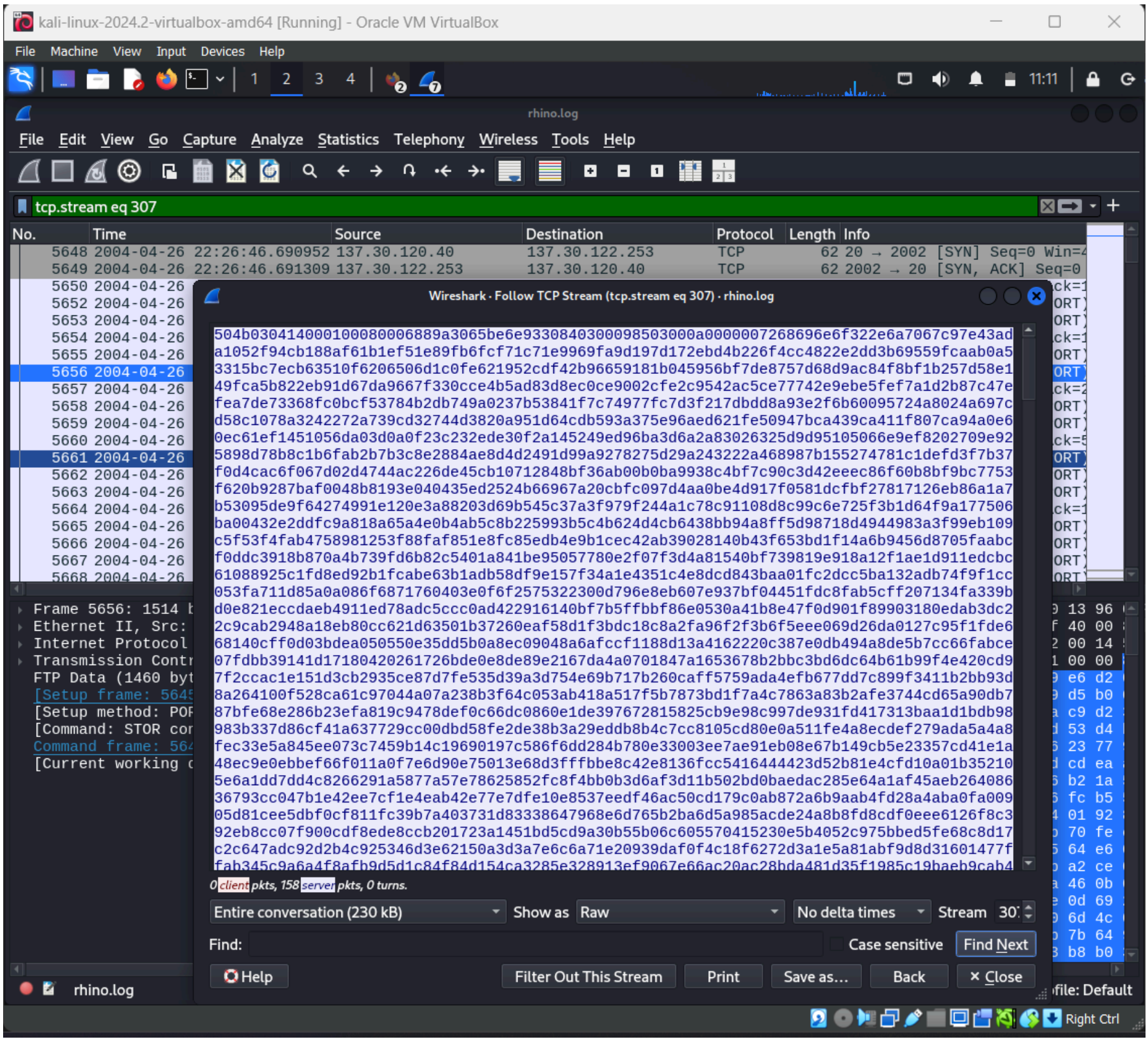
FTP Data (1460 bytes data)
[Setup frame: 5645]
[Setup method: PORT]
[Command: STOR contraband.zip]
Command frame: 5647
[Current working directory:]

Mark/Unmark Selected	Ctrl+M
Ignore/Unignore Selected	Ctrl+D
Set/Unset Time Reference	Ctrl+T
Time Shift...	Ctrl+Shift+T
Packet Comments	
Edit Resolved Name	
Apply as Filter	
Prepare as Filter	
Conversation Filter	
Colorize Conversation	
SCTP	
Follow	
Copy	
Protocol Preferences	
Decode As...	
Show Packet in New Window	

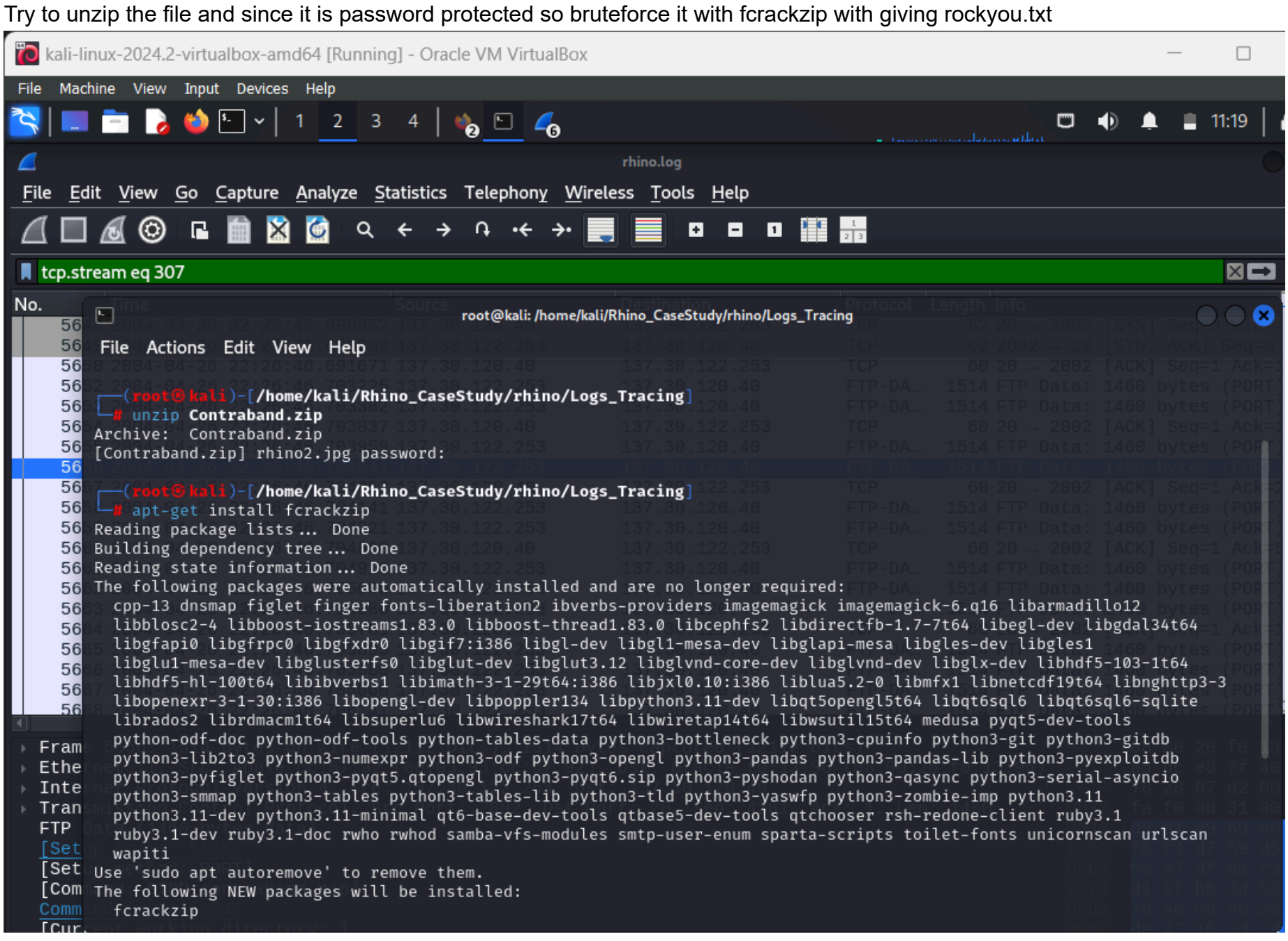
```
0000  08 00 20 f0 13 96
0010  05 dc e8 7f 40 00
0020  78 28 07 d2 00 14
0030  fa f0 0b 31 00 00
0040  dd 81 e9 b9 e6 d2
0050  f8 f4 d7 59 d5 b0
0060  5e a7 df 8a c9 d2
0070  d1 6f bb 3d 53 d4
0080  78 ae 08 56 23 77
0090  da 47 15 3d cd ea
00a0  53 c6 af 56 b2 1a
00b0  a3 6c d6 06 fc b5
00c0  e5 3b 41 94 01 92
00d0  4c 19 f3 5b 70 fe
00e0  8c 66 10 75 64 e6
00f0  43 17 88 8b a2 ce
0100  1d 43 49 4a 46 0b
0110  5f 84 fd be 0d 69
0120  b6 33 30 90 6d 4c
0130  d7 a9 4e 1b 7b 64
0140  ec d8 67 c3 b8 b0
```

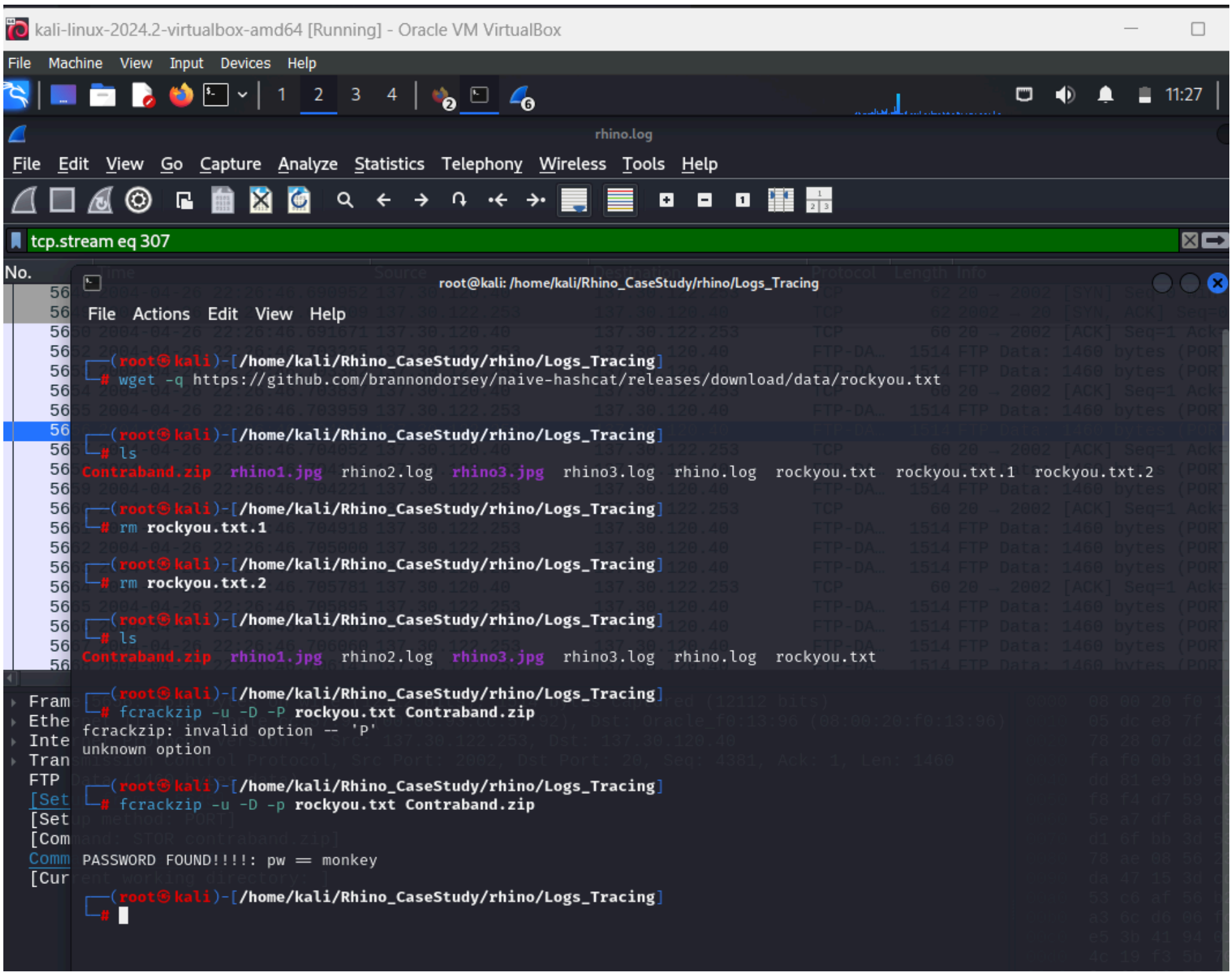
rhino.log Packets: 6557 Profile: Default

Right Ctrl

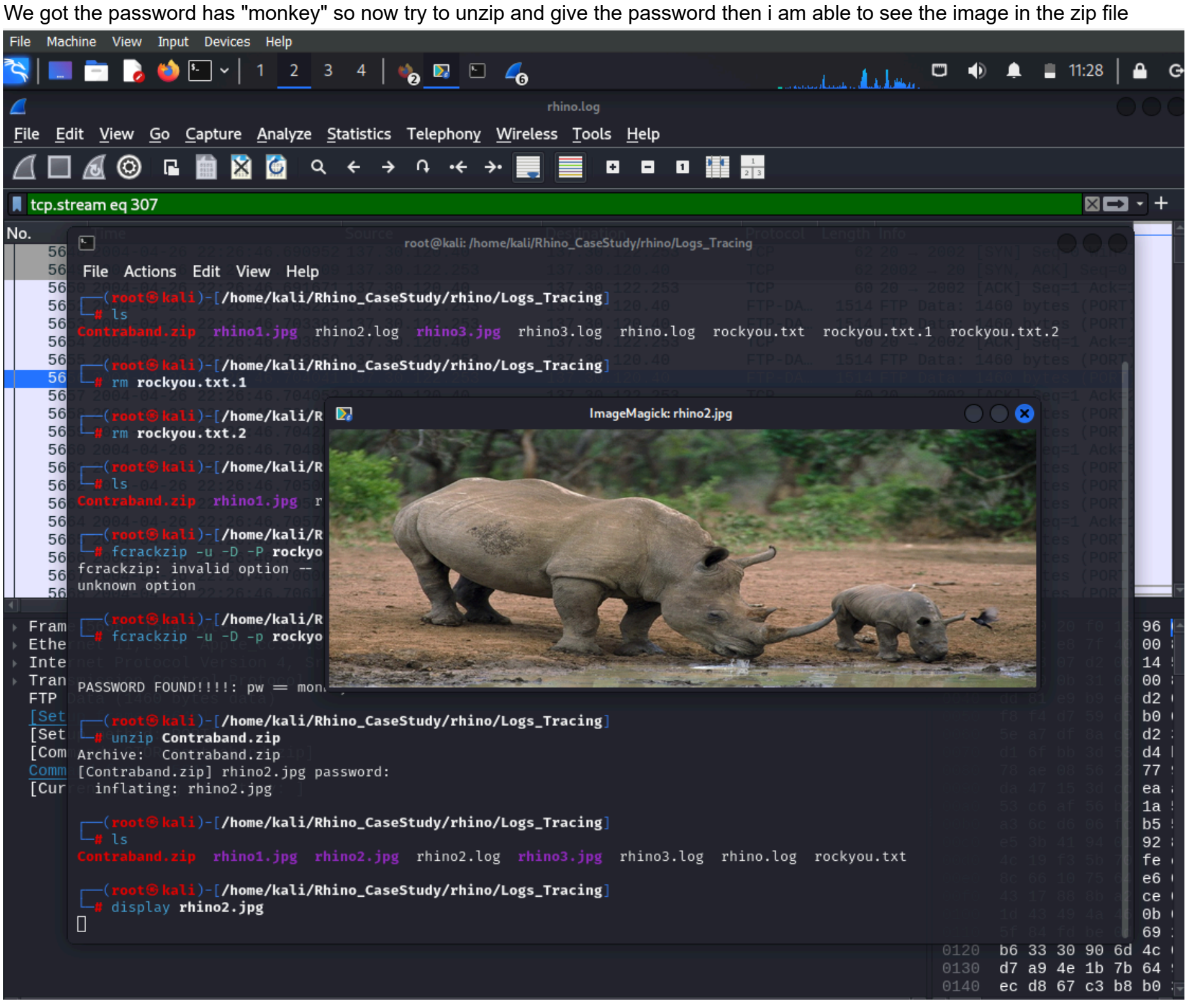


Try to unzip the file and since it is password protected so bruteforce it with fcrackzip with giving rockyou.txt



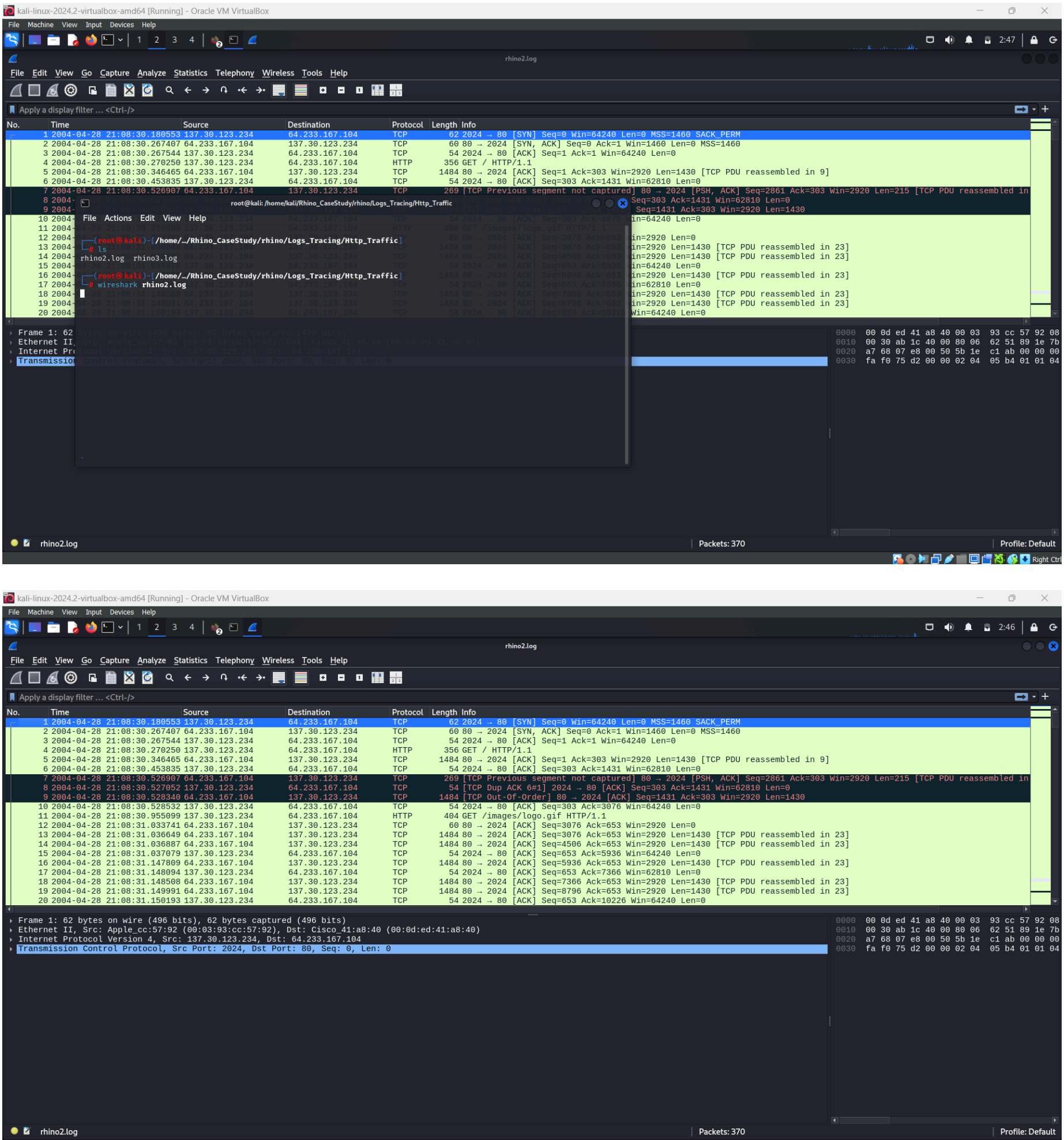


We got the password has "monkey" so now try to unzip and give the password then i am able to see the image in the zip file



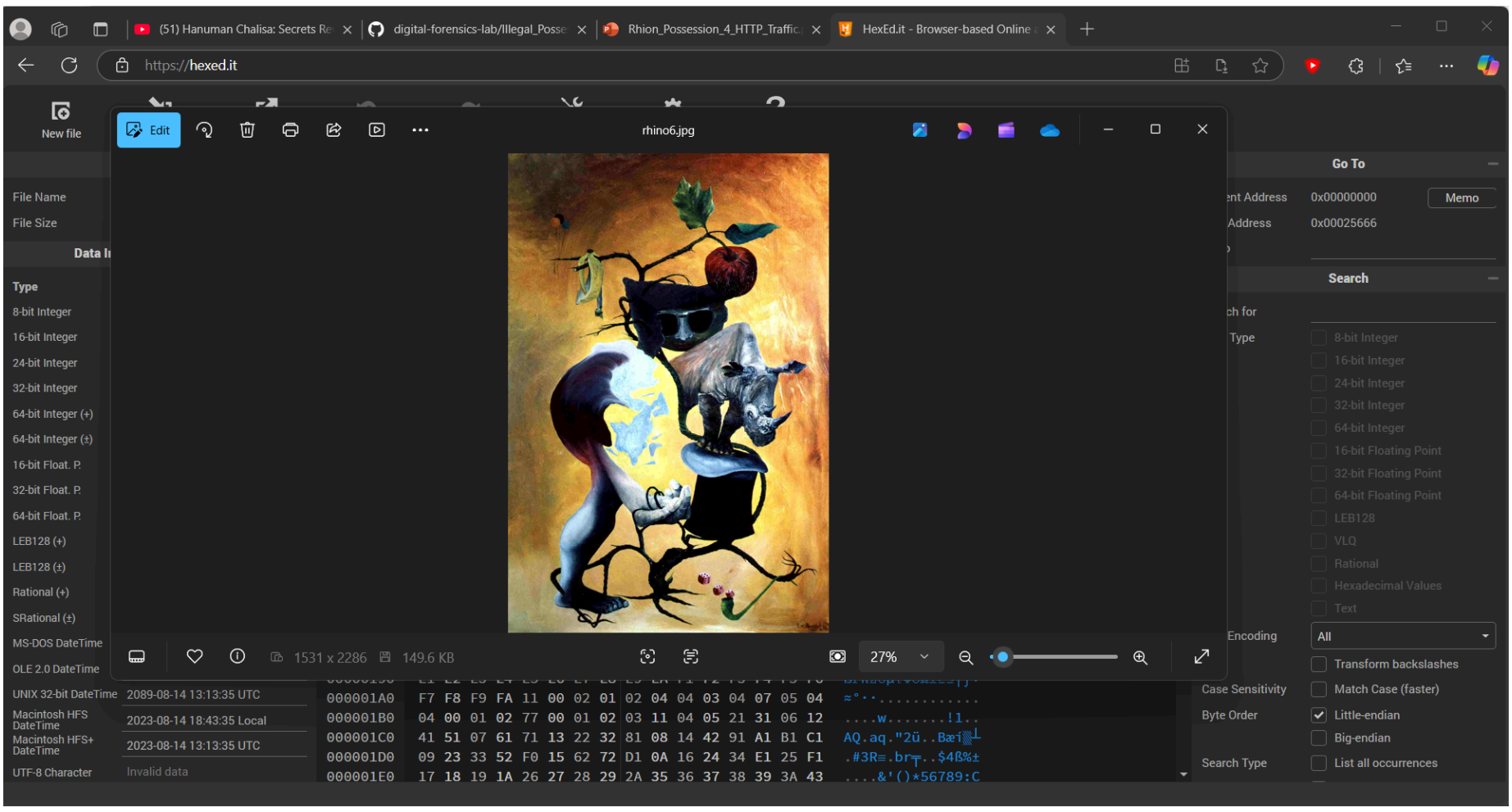
Examination Phase 2 : Tracing_Logs HTTP Traffic

At this phase we are provided with rhino2.log and need to analyze the http traffic

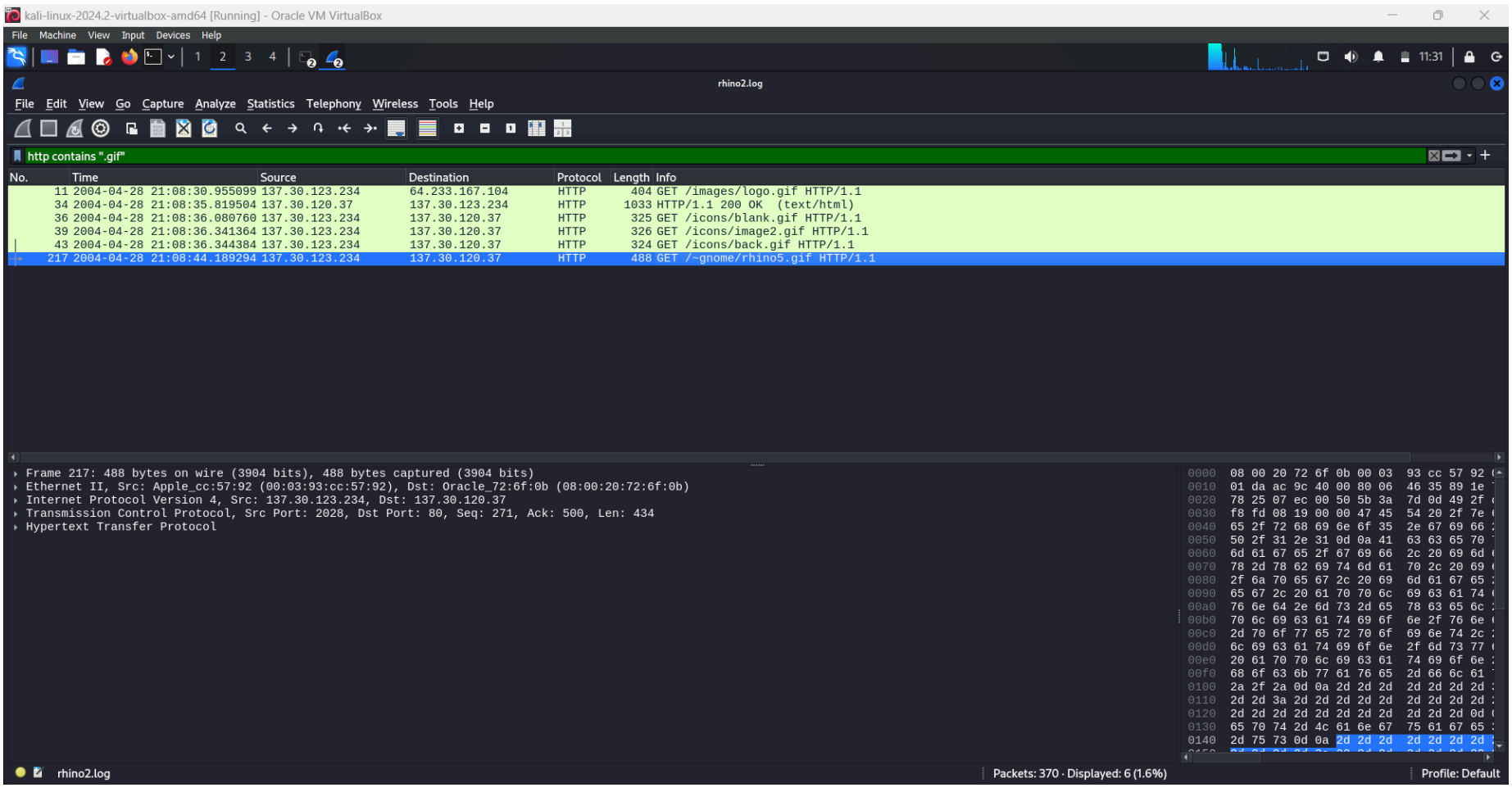


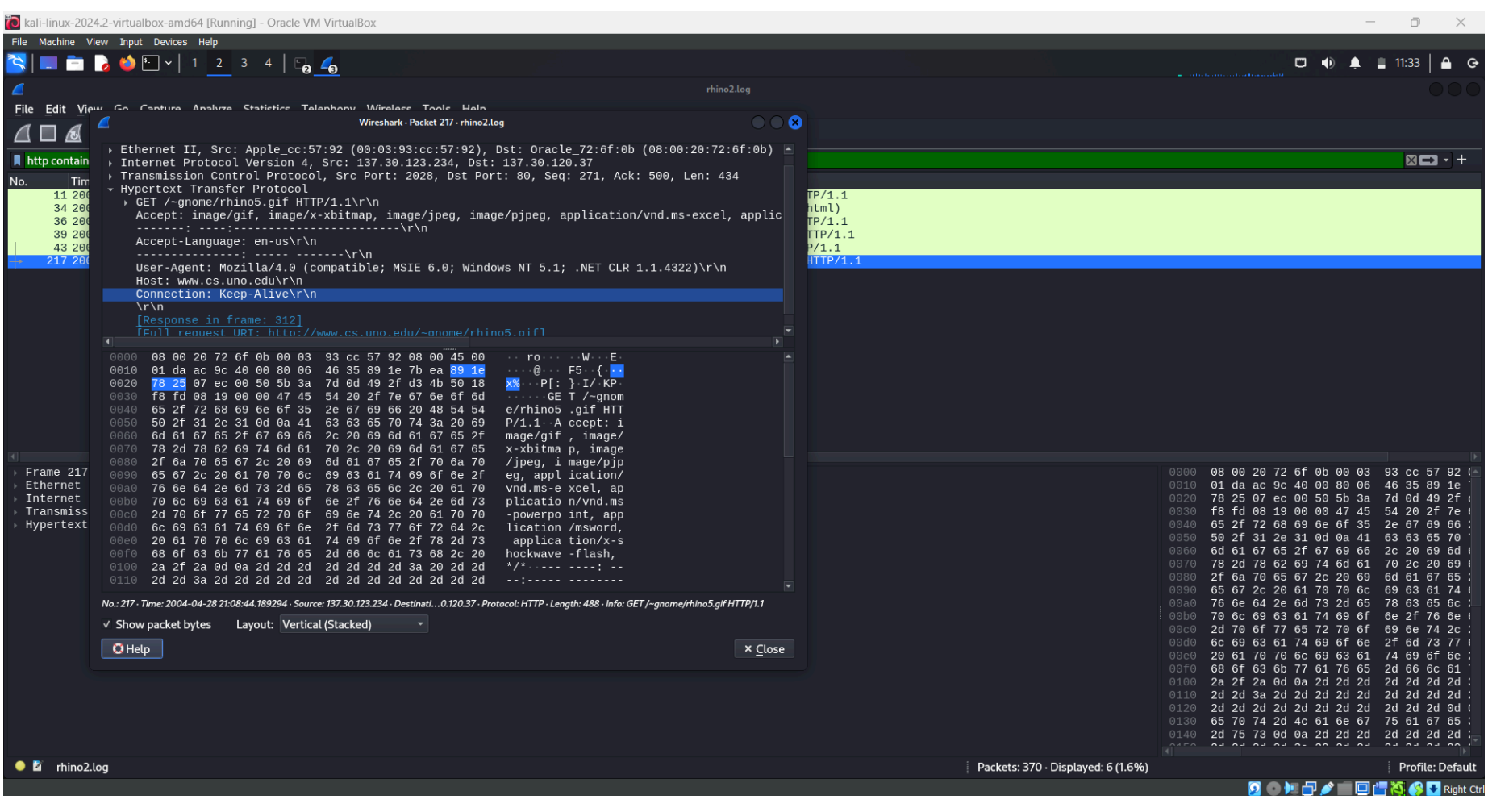
Trying by filtering for a jpg file by http contains ".jpg"

Finding out the "ffd8" to confirm the starting of image and ending "ffd9" copy and save in hex format and verify the magic number the image format and try to display the image i got the image .



Now try checking with filtering any ".gif" files which is http contains ".gif" download the packet file containing the gif and verify the magic number and we will able display it.





kali-linux-2024.2-virtualbox-amd64 [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

rhino2.log

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

http contains ".gif"

No.	Time	Source	Destination	Protocol	Length	Info
11	2004-04-28 21:08:30.955099	137.30.123.234	64.233.167.104	HTTP	404	GET /images/logo.gif HTTP/1.1
34	2004-04-28 21:08:35.819504	137.30.120.37	137.30.123.234	HTTP	1033	HTTP/1.1 200 OK (text/html)
36	2004-04-28 21:08:36.080760	137.30.123.234	137.30.120.37	HTTP	325	GET /icons/blank.gif HTTP/1.1
39	2004-04-28 21:08:36.341364	137.30.123.234	137.30.120.37	HTTP	326	GET /icons/image2.gif HTTP/1.1
43	2004-04-28 21:08:36.344384	137.30.123.234	137.30.120.37	HTTP	324	GET /icons/back.gif HTTP/1.1
217	2004-04-28 21:08:44.189294	137.30.123.234	137.30.120.37	HTTP	488	GET /~gnome/rhino5.gif HTTP/1.1

Mark/Unmark Selected Ctrl+M
Ignore/Unignore Selected Ctrl+D
Set/Unset Time Reference Ctrl+T
Time Shift... Ctrl+Shift+T
Packet Comments
Edit Resolved Name
Apply as Filter
Prepare as Filter
Conversation Filter
Colorize Conversation
SCTP
Follow
Copy
Protocol Preferences
Decode As...
Show Packet in New Window

Follow: HTTP Stream Ctrl+Alt+Shift+H; TCP Stream Ctrl+Alt+Shift+T

▸ Frame 217: 488 bytes on wire (3904 bits), 488 bytes cap
▸ Ethernet II, Src: Apple_cc:57:92 (00:03:93:cc:57:92), D
▸ Internet Protocol Version 4, Src: 137.30.123.234, Dst:
▸ Transmission Control Protocol, Src Port: 2028, Dst Port
▸ Hypertext Transfer Protocol

```
0000  08 00 20 72 6f 0b 00 03  93 cc 57 92
0010  01 da ac 9c 40 00 80 06  46 35 89 1e
0020  78 25 07 ec 00 50 5b 3a  7d 0d 49 2f
0030  f8 fd 08 19 00 00 47 45  54 20 2f 7e
0040  65 2f 72 68 69 6e 6f 35  2e 67 69 66
0050  50 2f 31 2e 31 0d 0a 41  63 63 65 70
0060  6d 61 67 65 2f 67 69 66  2c 20 69 6d
0070  78 2d 78 62 69 74 6d 61  70 2c 20 69
0080  2f 6a 70 65 67 2c 20 69  6d 61 67 65
0090  65 67 2c 20 61 70 70 6c  69 63 61 74
00a0  76 6e 64 2e 6d 73 2d 65  78 63 65 6c
00b0  70 6c 69 63 61 74 69 6f  6e 2f 76 6e
00c0  2d 70 6f 77 65 72 70 6f  69 6e 74 2c
00d0  6c 69 63 61 74 69 6f 6e  2f 6d 73 77
00e0  20 61 70 70 6c 69 63 61  74 69 6f 6e
00f0  68 6f 63 6b 77 61 76 65  2d 66 6c 61
0100  2a 2f 2a 0d 0a 2d 2d 2d  2d 2d 2d 2d
0110  2d 2d 3a 2d 2d 2d 2d 2d  2d 2d 2d 2d
0120  2d 2d 2d 2d 2d 2d 2d 2d  2d 2d 2d 0d
0130  65 70 74 2d 4c 61 6e 67  75 61 67 65
0140  2d 75 73 0d 0a 2d 2d 2d  2d 2d 2d 2d
```

rhino2.log | Packets: 370 · Displayed: 6 (1.6%) | Profile: Default

Right Ctrl

kali-linux-2024.2-virtualbox-amd64 [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

rhino2.log

File Edit View Go

Wireshark · Follow HTTP Stream (tcp.stream eq 2) · rhino2.log

tcp.stream eq 2

No. Time

```
Date: Wed, 28 Apr 2004 21:07:34 GMT
Server: Apache/1.3.29 (Unix)
Last-Modified: Wed, 28 Apr 2004 21:07:34 GMT
ETag: W/"19b1e7-14c91-40901d9c"
Accept-Ranges: bytes
Content-Length: 85137
Keep-Alive: timeout=15, max=99
Connection: Keep-Alive
Content-Type: image/gif

GIF89a.....
.....
.....IG8.....
.....njW.....
..........p.....!.....,.....@.....E>44...5....5...
....4.45.6n.66?cd5....V..&.....  ...     ....&.....&....  ...[.....doE.....5
6.?4.E?.5_.6k.Fu5>...n\4.4..`#A?.?...@.M....T.2..
#?.t.DcL
j.|dxV..:/.~.J......<..`#F..>.......t6:.a0.........oL.1.) .'..2K.|8..5....h88$...
..J..5......B.T..;.......o
2]<..+......).1.S..6..A\.q..d$O>u..g.d....e../d.!.T...J.2 5. ..j.R.i....|4.VH...cu..JHP.
..Z.L...k.u..L...k....>.r+K..X..".~.*9o. 6(d%C.
.h.(...)..$.W.L...:1...JH....n...)....I..Q..G. C.k,...[....u.a..v...:.$..2..p.W..1N..0d!
R....*.<p.......E.).....G...|...F&@Hr...
\.|..W.....*.BHV.8.....YC..$r."...

.].e~.   ..L....d|v....1.Qi5..Yi..X.....
>|..... ;       XA. .}4@.&......Q...  ....UPD8`M.    ..e7K+...At.........t.}.@t.x.C%?h..
7....K..D.:.zS..c.P.)1.0..S. ......)&.Z...0.......
^p..:.e..~E.V..5..E
 ......8...0% L.&...y.....E.1......Q..aD....A.3I..7...I7L..K...wm7.L.3..mY+C.P.s5be.....
@.Snp.@U..''<H..__h.s`
```

2 client pkts, 2 server pkts, 3 turns.

Entire conversation (86 kB) | Show as ASCII | No delta times | Stream 2

Find: | Case sensitive | Find Next

Help | Filter Out This Stream | Print | Save as... | Back | × Close

▸ Frame 46: 553 by
▸ Ethernet II, Src
▸ Internet Protoco
▸ Transmission Con
▸ Hypertext Transf
▸ Compuserve GIF,

kali-linux-2024.2-virtualbox-amd64 [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

rhino2.log

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

tcp.stream eq 2

No.	Time	Source	Destination	Protocol	Length	Info
299	2004-04-28 21:08:44.364944	137.30.123.234	137.30.120.37	TCP	54	2028 → 80 [ACK] Seq=705 Ack=73627 Win=64240 Len=0
300	2004-04-28 21:08:44.365930	137.30.120.37	137.30.123.234	TCP	1514	80 → 2028 [PSH, ACK] Seq=73627 Ack=705 Win=49206 Len=1460 [TCP PDU reassembled in 312]
301	2004-04-28 21:08:44.367391	137.30.120.37	137.30.123.234	TCP	1514	80 → 2028 [ACK] Seq=75087 Ack=705 Win=49206 Len=1460 [TCP PDU reassembled in 312]
302	2004-04-28 21:08:44.367600	137.30.123.234	137.30.120.37	TCP	54	2028 → 80 [ACK] Seq=705 Ack=76547 Win=64240 Len=0
303	2004-04-28 21:08:44.368605	137.30.120.37	137.30.123.234	TCP	1514	80 → 2028 [PSH, ACK] Seq=76547 Ack=705 Win=49206 Len=1460 [TCP PDU reassembled in 312]
304	2004-04-28 21:08:44.371068	137.30.120.37	137.30.123.234	TCP	1514	80 → 2028 [ACK] Seq=78007 Ack=705 Win=49206 Len=1460 [TCP PDU reassembled in 312]
305	2004-04-28 21:08:44.371291	137.30.123.234	137.30.120.37	TCP	54	2028 → 80 [ACK] Seq=705 Ack=79467 Win=64240 Len=0
306	2004-04-28 21:08:44.372248	137.30.120.37	137.30.123.234	TCP	1514	80 → 2028 [PSH, ACK] Seq=79467 Ack=705 Win=49206 Len=1460 [TCP PDU reassembled in 312]
307	2004-04-28 21:08:44.374970	137.30.120.37	137.30.123.234	TCP	1514	80 → 2028 [ACK] Seq=80927 Ack=705 Win=49206 Len=1460 [TCP PDU reassembled in 312]
308	2004-04-28 21:08:44.375202	137.30.123.234	137.30.120.37	TCP	54	2028 → 80 [ACK] Seq=705 Ack=82387 Win=64240 Len=0
309	2004-04-28 21:08:44.376142	137.30.120.37	137.30.123.234	TCP	1514	80 → 2028 [PSH, ACK] Seq=82387 Ack=705 Win=49206 Len=1460 [TCP PDU reassembled in 312]
310	2004-04-28 21:08:44.377606	137.30.120.37	137.30.123.234	TCP	1514	80 → 2028 [ACK] Seq=83847 Ack=705 Win=49206 Len=1460 [TCP PDU reassembled in 312]
311	2004-04-28 21:08:44.377797	137.30.123.234	137.30.120.37	TCP	54	2028 → 80 [ACK] Seq=705 Ack=85307 Win=64240 Len=0
312	2004-04-28 21:08:44.378087	137.30.120.37	137.30.123.234	HTTP	675	HTTP/1.1 200 OK (GIF89a)
313	2004-04-28 21:08:44.498636	137.30.123.234	137.30.120.37	TCP	54	2028 → 80 [ACK] Seq=705 Ack=85928 Win=63619 Len=0
320	2004-04-28 21:08:53.574938	137.30.123.234	137.30.120.37	TCP	54	2028 → 80 [FIN, ACK] Seq=705 Ack=85928 Win=63619 Len=0
323	2004-04-28 21:08:53.575584	137.30.120.37	137.30.123.234	TCP	60	80 → 2028 [ACK] Seq=85928 Ack=706 Win=49206 Len=0
326	2004-04-28 21:08:53.576313	137.30.120.37	137.30.123.234	TCP	60	80 → 2028 [FIN, ACK] Seq=85928 Ack=706 Win=49206 Len=0
327	2004-04-28 21:08:53.576343	137.30.123.234	137.30.120.37	TCP	54	2028 → 80 [ACK] Seq=706 Ack=85929 Win=63619 Len=0

kali-linux-2024.2-virtualbox-amd64 [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

rhino2.log

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

tcp.stream eq 2

No.	Time	Source	Destination	Protocol	Length	Info
299	2004-04-28 21:08:44.364944	137.30.123.234	137.30.120.37	TCP	54	2028 → 80 [ACK] Seq=705 Ack=73627 Win=64240 Len=0
300	2004-04-28 21:08:44.365930	137.30.120.37	137.30.123.234	TCP	1514	80 → 2028 [PSH, ACK] Seq=73627 Ack=705 Win=49206 Len=1460 [TCP PDU reassembled in 312]
301	2004-04-28 21:08:44.367391	137.30.120.37	137.30.123.234	TCP	1514	80 → 2028 [ACK] Seq=75087 Ack=705 Win=49206 Len=1460 [TCP PDU reassembled in 312]
302	2004-04-28 21:08:44.367600	137.30.123.234	137.30.120.37	TCP	54	2028 → 80 [ACK] Seq=705 Ack=76547 Win=64240 Len=0
303	2004-04-28 21:08:44.368605	137.30.120.37	137.30.123.234	TCP	1514	80 → 2028 [PSH, ACK] Seq=76547 Ack=705 Win=49206 Len=1460 [TCP PDU reassembled in 312]
304	2004-04-28 21:08:44.371068	137.30.120.37	137.30.123.234	TCP	1514	80 → 2028 [ACK] Seq=78007 Ack=705 Win=49206 Len=1460 [TCP PDU reassembled in 312]
305	2004-04-28 21:08:44.371291	137.30.123.234	137.30.120.37	TCP	54	2028 → 80 [ACK] Seq=705 Ack=79467 Win=64240 Len=0
306	2004-04-28 21:08:44.372248	137.30.120.37	137.30.123.234	TCP	1514	80 → 2028 [PSH, ACK] Seq=79467 Ack=705 Win=49206 Len=1460 [TCP PDU reassembled in 312]
307	2004-04-28 21:08:44.374970	137.30.120.37	137.30.123.234	TCP	1514	80 → 2028 [ACK] Seq=80927 Ack=705 Win=49206 Len=1460 [TCP PDU reassembled in 312]
308	2004-04-28 21:08:44.375202	137.30.123.234	137.30.120.37	TCP	54	2028 → 80 [ACK] Seq=705 Ack=82387 Win=64240 Len=0
309	2004-04-28 21:08:44.376142	137.30.120.37	137.30.123.234	TCP	1514	80 → 2028 [PSH, ACK] Seq=82387 Ack=705 Win=49206 Len=1460 [TCP PDU reassembled in 312]
310	2004-04-28 21:08:44.377606	137.30.120.37	137.30.123.234	TCP	1514	80 → 2028 [ACK] Seq=83847 Ack=705 Win=49206 Len=1460 [TCP PDU reassembled in 312]
311	2004-04-28 21:08:44.377797	137.30.123.234	137.30.120.37	TCP	54	2028 → 80 [ACK] Seq=705 Ack=85307 Win=64240 Len=0
312	2004-04-28 21:08:44.378087	137.30.120.37	137.30.123.234	HTTP	675	HTTP/1.1 200 OK (GIF89a)
313	2004-04-28 21:08:44.498636	137.30.123.234	137.30.120.37	TCP	54	2028 → 80 [ACK] Seq=705 Ack=85928 Win=63619 Len=0
320	2004-04-28 21:08:53.574938	137.30.123.234	137.30.120.37	TCP	54	2028 → 80 [FIN, ACK] Seq=705 Ack=85928 Win=63619 Len=0
323	2004-04-28 21:08:53.575584	137.30.120.37	137.30.123.234	TCP	60	80 → 2028 [ACK] Seq=85928 Ack=706 Win=49206 Len=0
326	2004-04-28 21:08:53.576313	137.30.120.37	137.30.123.234	TCP	60	80 → 2028 [FIN, ACK] Seq=85928 Ack=706 Win=49206 Len=0
327	2004-04-28 21:08:53.576343	137.30.123.234	137.30.120.37	TCP	54	2028 → 80 [ACK] Seq=706 Ack=85929 Win=63619 Len=0

```
▸ Frame 312: 675 bytes on wire (5400 bits), 675 bytes captured (5400 bits)
▸ Ethernet II, Src: Oracle_72:6f:0b (08:00:20:72:6f:0b), Dst: Apple_cc:57:92 (00:03:93:cc:57:92)
▸ Internet Protocol Version 4, Src: 137.30.120.37, Dst: 137.30.123.234
▸ Transmission Control Protocol, Src Port: 80, Dst Port: 2028, Seq: 85307, Ack: 705, Len: 621
▸ [ …]60 Reassembled TCP Segments (85428 bytes): #219(1460), #220(1460), #222(1460), #223(1460), #225(1460), #226(1460), #228(1460), #229(1460), #231(1460), #232(14
▸ Hypertext Transfer Protocol
▾ Compuserve GIF, Version: GIF89a
    Version: GIF89a
    Screen width: 400
    Screen height: 275
  ▸ Global settings: (Global color table present) (7 bits per color) (7 bits per pixel)
    Background color index: 0
    Global color map […]: 000000ffffffbfdf2fbfdeef7f9eae7e9d6eff1dfebeddbfbfdebf9fbe9ebeddde7e9d9f9fbebfbfde7f9fbe5f9fbe7dfe0cffdfde7fbfbe5d3d3c1fbfbe7f9f9e5e9e9d7c
  ▸ Extension: Graphics Control
  ▸ Image
    Trailer (End of the GIF stream)
```

```
0000  00 03 93 cc 57 92 08 00  20 72 6f 0b
0010  02 95 b1 f9 40 00 40 06  80 1d 89 1e
0020  7b ea 00 50 07 ec 49 31  1e 92 5b 3a
0030  c0 36 f2 8e 00 00 ce 44  dc cb 53 14
0040  72 85 f3 8d 5b a6 46 bc  64 bb 43 a2
0050  59 85 d5 53 40 90 95 d4  57 17 c9 10
0060  b9 31 50 65 c0 00 83 60  8d f5 04 83
0070  94 65 68 b5 bb d6 1b 98  dd 3b 80 68
0080  02 35 20 90 41 02 dd 26  42 cc 10 3f
0090  da 64 53 95 24 cf 38 e6  55 2e 4e aa
00a0  68 50 c4 80 0f 20 24 c1  11 b6 0c a8
00b0  13 0e ff 2e e2 0c 81 cb  c3 76 7c 14
00c0  db 72 7b 6b 20 25 8d 45  5f b8 d1 cf
00d0  a0 48 d9 84 f3 cc bc cd  6d d9 ea 80
00e0  0d 12 6c e2 da 6d 08 73  21 e4 10 5b
00f0  5f f0 b5 84 3e 47 2d c5  92 6f 17 cd
0100  d0 9c b0 77 32 65 26 ed  57 cd 90 b4
0110  03 ce 89 9d 04 c5 d7 36  1e 7f 55 46
0120  98 c7 c2 b5 6b 15 e4 4e  79 82 a2 58
0130  bc 72 83 9a 8f 54 5c 70  d1 59 d6 9d
```

Frame (675 bytes) | Reassembled TCP (85428 bytes)

kali-linux-2024.2-virtualbox-amd64 [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

rhino2.log

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

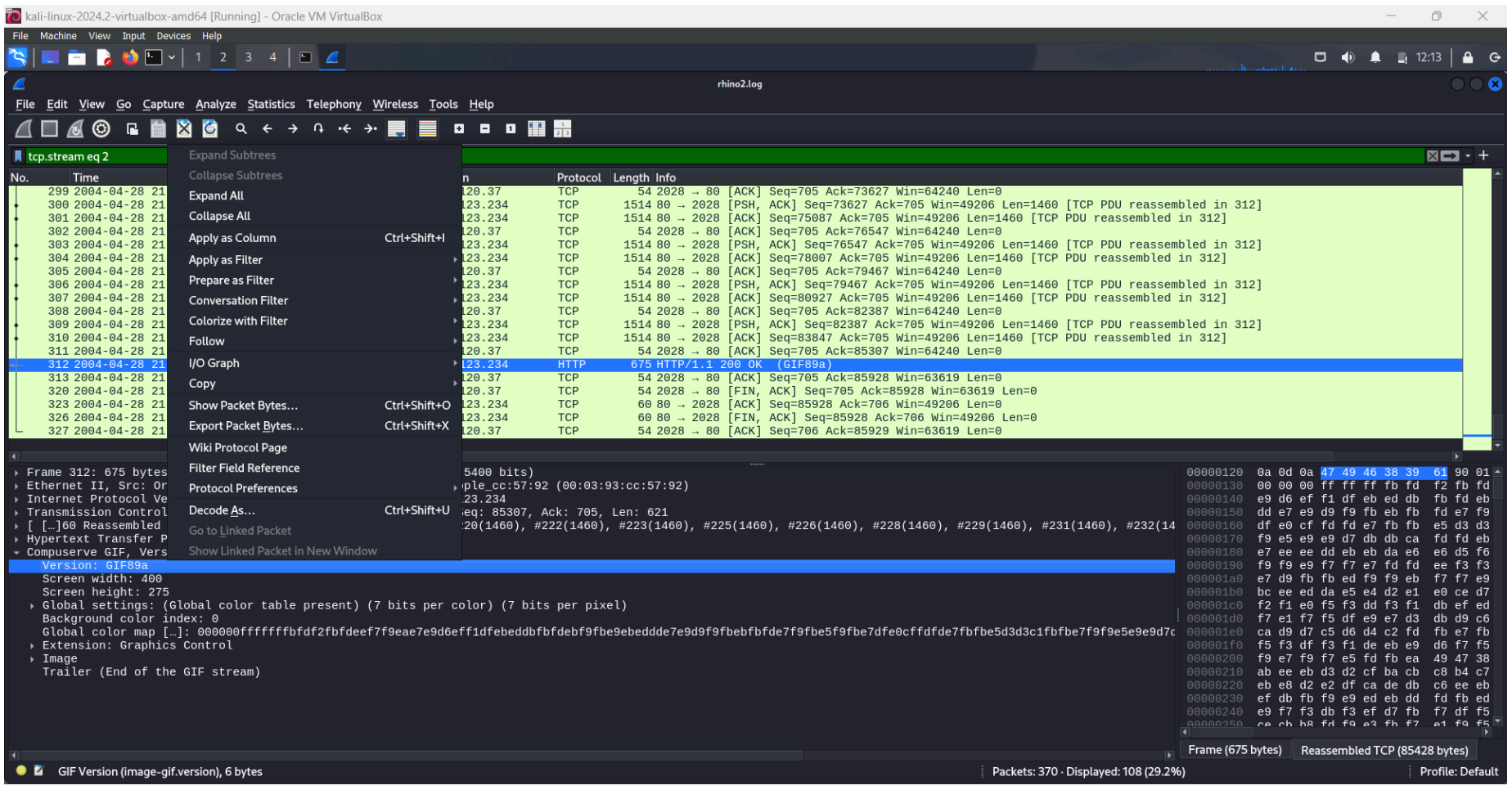
tcp.stream eq 2

No.	Time	Source	Destination	Protocol	Length	Info
299	2004-04-28 21:08:44.364944	137.30.123.234	137.30.120.37	TCP	54	2028 → 80 [ACK] Seq=705 Ack=73627 Win=64240 Len=0
300	2004-04-28 21:08:44.365930	137.30.120.37	137.30.123.234	TCP	1514	80 → 2028 [PSH, ACK] Seq=73627 Ack=705 Win=49206 Len=1460 [TCP PDU reassembled in 312]
301	2004-04-28 21:08:44.367391	137.30.120.37	137.30.123.234	TCP	1514	80 → 2028 [ACK] Seq=75087 Ack=705 Win=49206 Len=1460 [TCP PDU reassembled in 312]
302	2004-04-28 21:08:44.367600	137.30.123.234	137.30.120.37	TCP	54	2028 → 80 [ACK] Seq=705 Ack=76547 Win=64240 Len=0
303	2004-04-28 21:08:44.368605	137.30.120.37	137.30.123.234	TCP	1514	80 → 2028 [PSH, ACK] Seq=76547 Ack=705 Win=49206 Len=1460 [TCP PDU reassembled in 312]
304	2004-04-28 21:08:44.371068	137.30.120.37	137.30.123.234	TCP	1514	80 → 2028 [ACK] Seq=78007 Ack=705 Win=49206 Len=1460 [TCP PDU reassembled in 312]
305	2004-04-28 21:08:44.371291	137.30.123.234	137.30.120.37	TCP	54	2028 → 80 [ACK] Seq=705 Ack=79467 Win=64240 Len=0
306	2004-04-28 21:08:44.372248	137.30.120.37	137.30.123.234	TCP	1514	80 → 2028 [PSH, ACK] Seq=79467 Ack=705 Win=49206 Len=1460 [TCP PDU reassembled in 312]
307	2004-04-28 21:08:44.374970	137.30.120.37	137.30.123.234	TCP	1514	80 → 2028 [ACK] Seq=80927 Ack=705 Win=49206 Len=1460 [TCP PDU reassembled in 312]
308	2004-04-28 21:08:44.375202	137.30.123.234	137.30.120.37	TCP	54	2028 → 80 [ACK] Seq=705 Ack=82387 Win=64240 Len=0
309	2004-04-28 21:08:44.376142	137.30.120.37	137.30.123.234	TCP	1514	80 → 2028 [PSH, ACK] Seq=82387 Ack=705 Win=49206 Len=1460 [TCP PDU reassembled in 312]
310	2004-04-28 21:08:44.377606	137.30.120.37	137.30.123.234	TCP	1514	80 → 2028 [ACK] Seq=83847 Ack=705 Win=49206 Len=1460 [TCP PDU reassembled in 312]
311	2004-04-28 21:08:44.377797	137.30.123.234	137.30.120.37	TCP	54	2028 → 80 [ACK] Seq=705 Ack=85307 Win=64240 Len=0
312	2004-04-28 21:08:44.378087	137.30.120.37	137.30.123.234	HTTP	675	HTTP/1.1 200 OK (GIF89a)
313	2004-04-28 21:08:44.498636	137.30.123.234	137.30.120.37	TCP	54	2028 → 80 [ACK] Seq=705 Ack=85928 Win=63619 Len=0
320	2004-04-28 21:08:53.574938	137.30.123.234	137.30.120.37	TCP	54	2028 → 80 [FIN, ACK] Seq=705 Ack=85928 Win=63619 Len=0
323	2004-04-28 21:08:53.575584	137.30.120.37	137.30.123.234	TCP	60	80 → 2028 [ACK] Seq=85928 Ack=706 Win=49206 Len=0
326	2004-04-28 21:08:53.576313	137.30.120.37	137.30.123.234	TCP	60	80 → 2028 [FIN, ACK] Seq=85928 Ack=706 Win=49206 Len=0
327	2004-04-28 21:08:53.576343	137.30.123.234	137.30.120.37	TCP	54	2028 → 80 [ACK] Seq=706 Ack=85929 Win=63619 Len=0

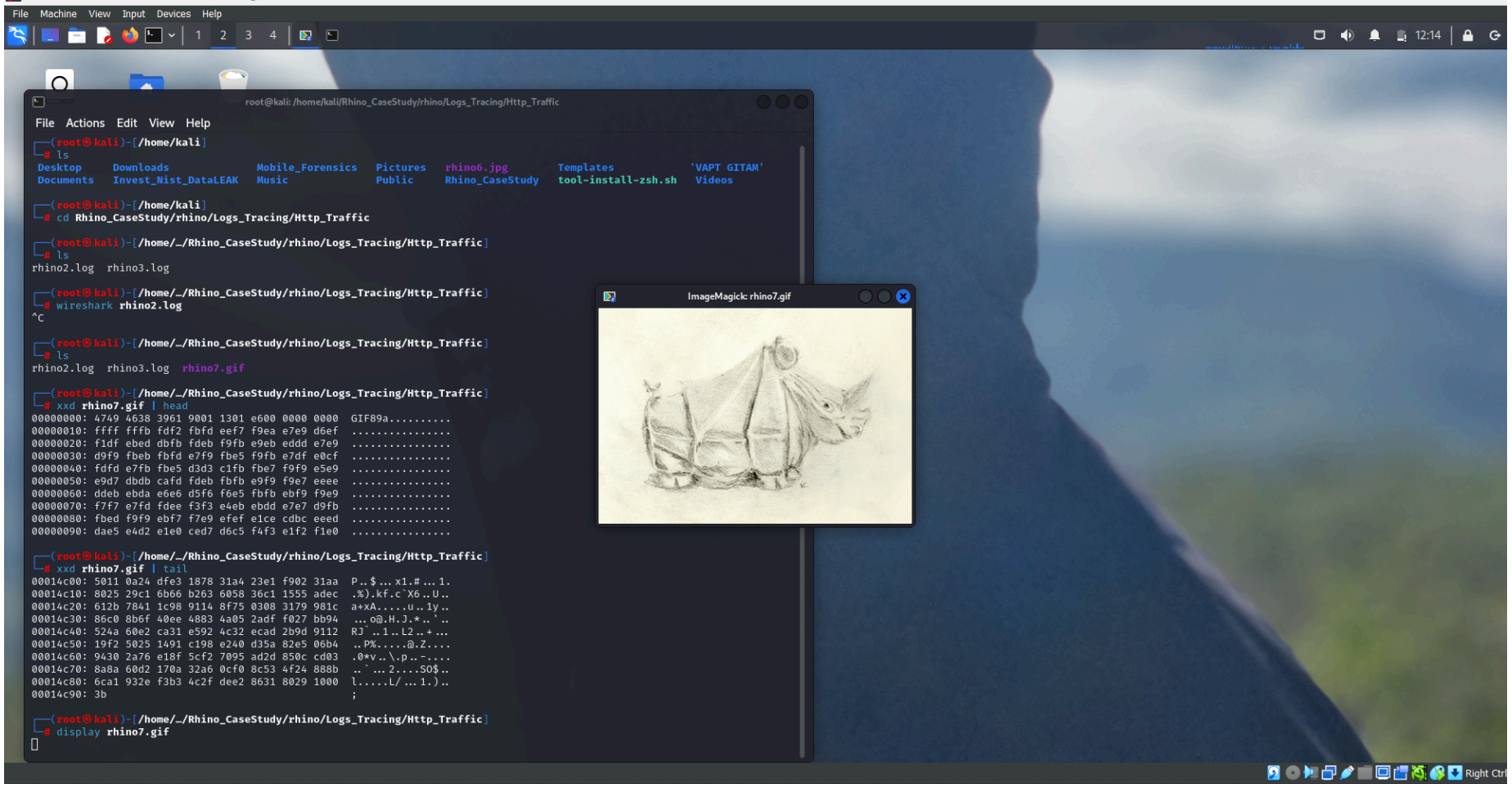
```
▸ Internet Protocol Version 4, Src: 137.30.120.37, Dst: 137.30.123.234
▸ Transmission Control Protocol, Src Port: 80, Dst Port: 2028, Seq: 85307, Ack: 705, Len: 621
▸ [ […]60 Reassembled TCP Segments (85428 bytes): #219(1460), #220(1460), #222(1460), #223(1460), #225(1460), #226(1460), #228(1460), #229(1460), #231(1460), #232(
▾ Hypertext Transfer Protocol
  ▸ HTTP/1.1 200 OK\r\n
    Date: Wed, 28 Apr 2004 21:07:34 GMT\r\n
    Server: Apache/1.3.29 (Unix)\r\n
    Last-Modified: Wed, 28 Apr 2004 21:07:34 GMT\r\n
    ETag: W/"19b1e7-14c91-40901d9c"\r\n
    Accept-Ranges: bytes\r\n
  ▾ Content-Length: 85137\r\n
      [Content length: 85137]
    Keep-Alive: timeout=15, max=99\r\n
    Connection: Keep-Alive\r\n
    Content-Type: image/gif\r\n
    \r\n
    [Request in frame: 217]
    [Time since request: 0.188793000 seconds]
    [Request URI: /~gnome/rhino5.gif]
    [Full request URI: http://www.cs.uno.edu/~gnome/rhino5.gif]
    File Data: 85137 bytes
```

0000 00 03 93 cc 57 92 08 00 20 72 6f 0b
0010 02 95 b1 f9 40 00 40 06 80 1d 89 1e
0020 7b ea 00 50 07 ec 49 31 1e 92 5b 3a
0030 c0 36 f2 8e 00 00 ce 44 dc cb 53 14
0040 72 85 f3 8d 5b a6 46 bc 64 bb 43 a2
0050 59 85 d5 53 40 90 95 d4 57 17 c9 10
0060 b9 31 50 65 c0 00 83 60 8d f5 04 83
0070 94 65 68 b5 bb d6 1b 98 dd 3b 80 68
0080 02 35 20 90 41 02 dd 26 42 cc 10 3f
0090 da 64 53 95 24 cf 38 e6 55 2e 4e aa
00a0 68 50 c4 80 0f 20 24 c1 11 b6 0c a8
00b0 13 0e ff 2e e2 0c 81 cb c3 76 7c 14
00c0 db 72 7b 6b 20 25 8d 45 5f b8 d1 cf
00d0 a0 48 d9 84 f3 cc bc cd 6d d9 ea 80
00e0 0d 12 6c e2 da 6d 08 73 21 e4 10 5b
00f0 5f f0 b5 84 3e 47 2d c5 92 6f 17 cd
0100 d0 9c b0 77 32 65 26 ed 57 cd 90 b4
0110 03 ce 89 9d 04 c5 d7 36 1e 7f 55 46
0120 98 c7 c2 b5 6b 15 e4 4e 79 82 a2 58
0130 bc 72 83 9a 8f 54 5c 70 d1 59 d6 9d

Frame (675 bytes) | Reassembled TCP (85428 bytes)



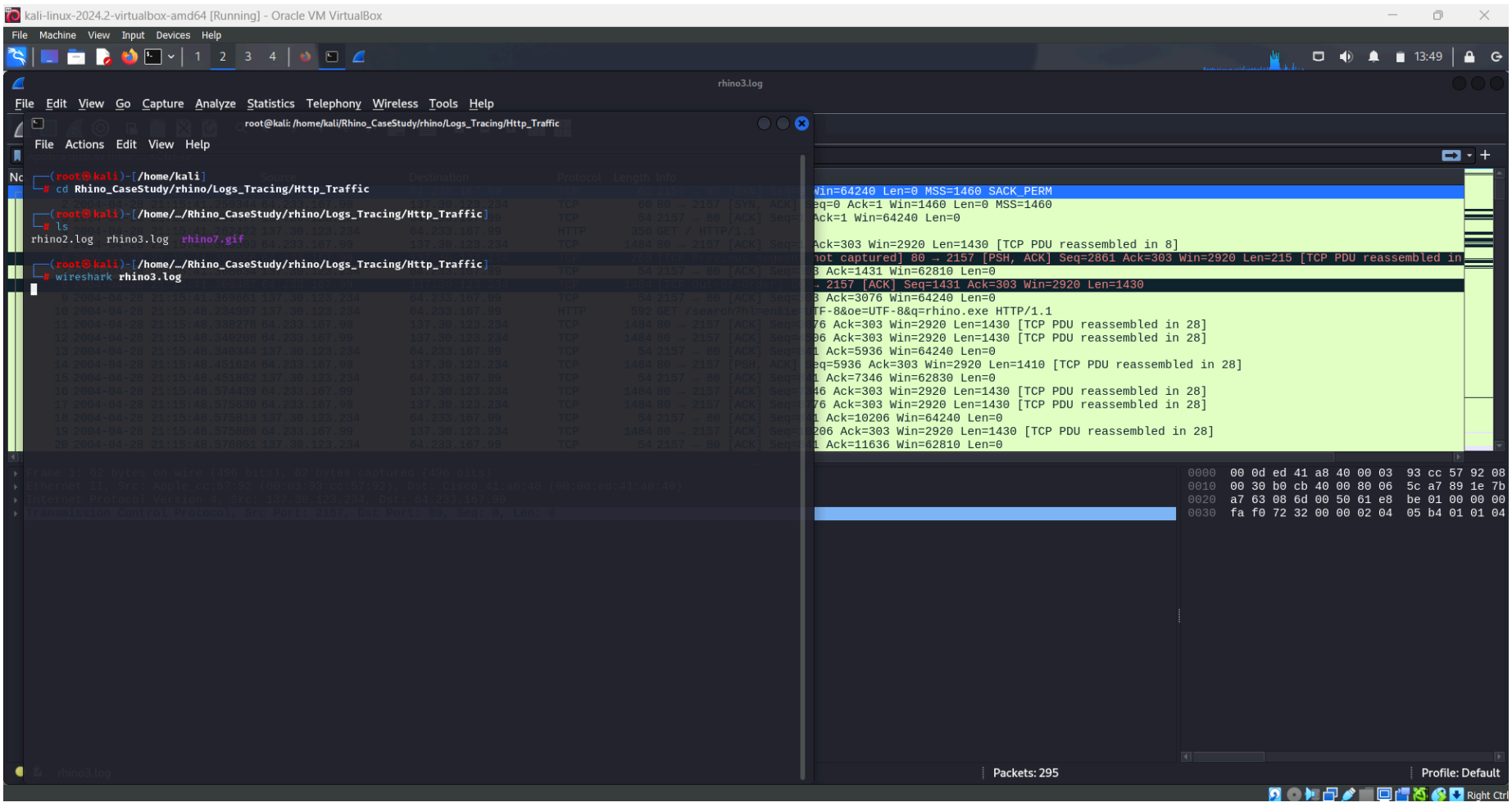

```
┌──(root㉿kali)-[/home/…/Rhino_CaseStudy/rhino/Logs_Tracing/Http_Traffic]
└─# xxd rhino7.gif | head
00000000: 4749 4638 3961 9001 1301 e600 0000 0000  GIF89a..........
00000010: ffff fffb fdf2 fbfd eef7 f9ea e7e9 d6ef  ................
00000020: f1df ebed dbfb fdeb f9fb e9eb eddd e7e9  ................
00000030: d9f9 fbeb fbfd e7f9 fbe5 f9fb e7df e0cf  ................
00000040: fdfd e7fb fbe5 d3d3 c1fb fbe7 f9f9 e5e9  ................
00000050: e9d7 dbdb cafd fdeb fbfb e9f9 f9e7 eeee  ................
00000060: ddeb ebda e6e6 d5f6 f6e5 fbfb ebf9 f9e9  ................
00000070: f7f7 e7fd fdee f3f3 e4eb ebdd e7e7 d9fb  ................
00000080: fbed f9f9 ebf7 f7e9 efef e1ce cdbc eeed  ................
00000090: dae5 e4d2 e1e0 ced7 d6c5 f4f3 e1f2 f1e0  ................

┌──(root㉿kali)-[/home/…/Rhino_CaseStudy/rhino/Logs_Tracing/Http_Traffic]
└─# xxd rhino7.gif | tail
00014c00: 5011 0a24 dfe3 1878 31a4 23e1 f902 31aa  P..$...x1.#...1.
00014c10: 8025 29c1 6b66 b263 6058 36c1 1555 adec  .%).kf.c`X6..U..
00014c20: 612b 7841 1c98 9114 8f75 0308 3179 981c  a+xA.....u..1y..
00014c30: 86c0 8b6f 40ee 4883 4a05 2adf f027 bb94  ...o@.H.J.*..'..
00014c40: 524a 60e2 ca31 e592 4c32 ecad 2b9d 9112  RJ`..1..L2..+...
00014c50: 19f2 5025 1491 c198 e240 d35a 82e5 06b4  ..P%.....@.Z....
00014c60: 9430 2a76 e18f 5cf2 7095 ad2d 850c cd03  .0*v..\.p..-....
00014c70: 8a8a 60d2 170a 32a6 0cf0 8c53 4f24 888b  ..`...2....SO$..
00014c80: 6ca1 932e f3b3 4c2f dee2 8631 8029 1000  l.....L/...1.)..
00014c90: 3b                                       ;

┌──(root㉿kali)-[/home/…/Rhino_CaseStudy/rhino/Logs_Tracing/Http_Traffic]
└─#
```

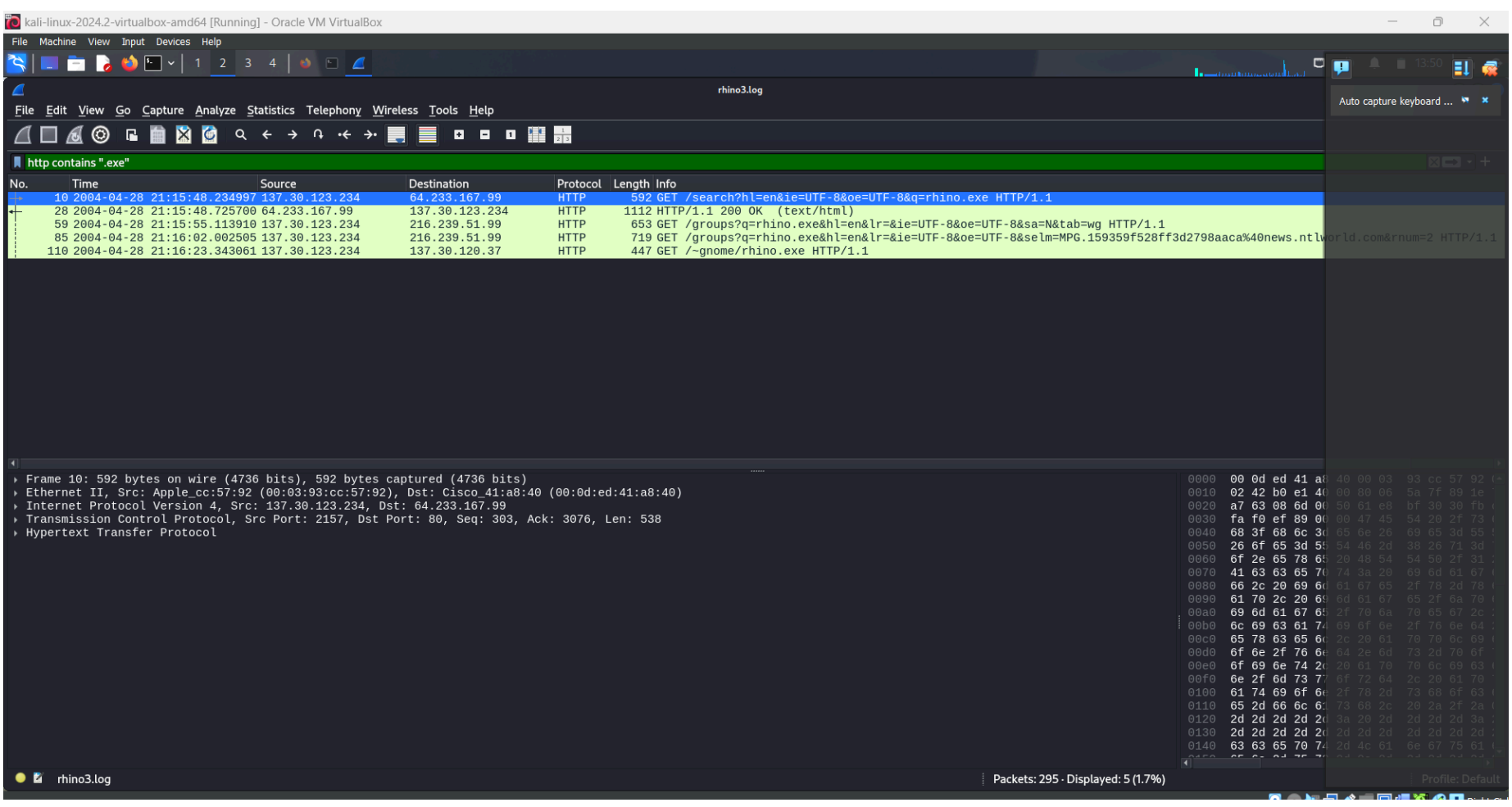


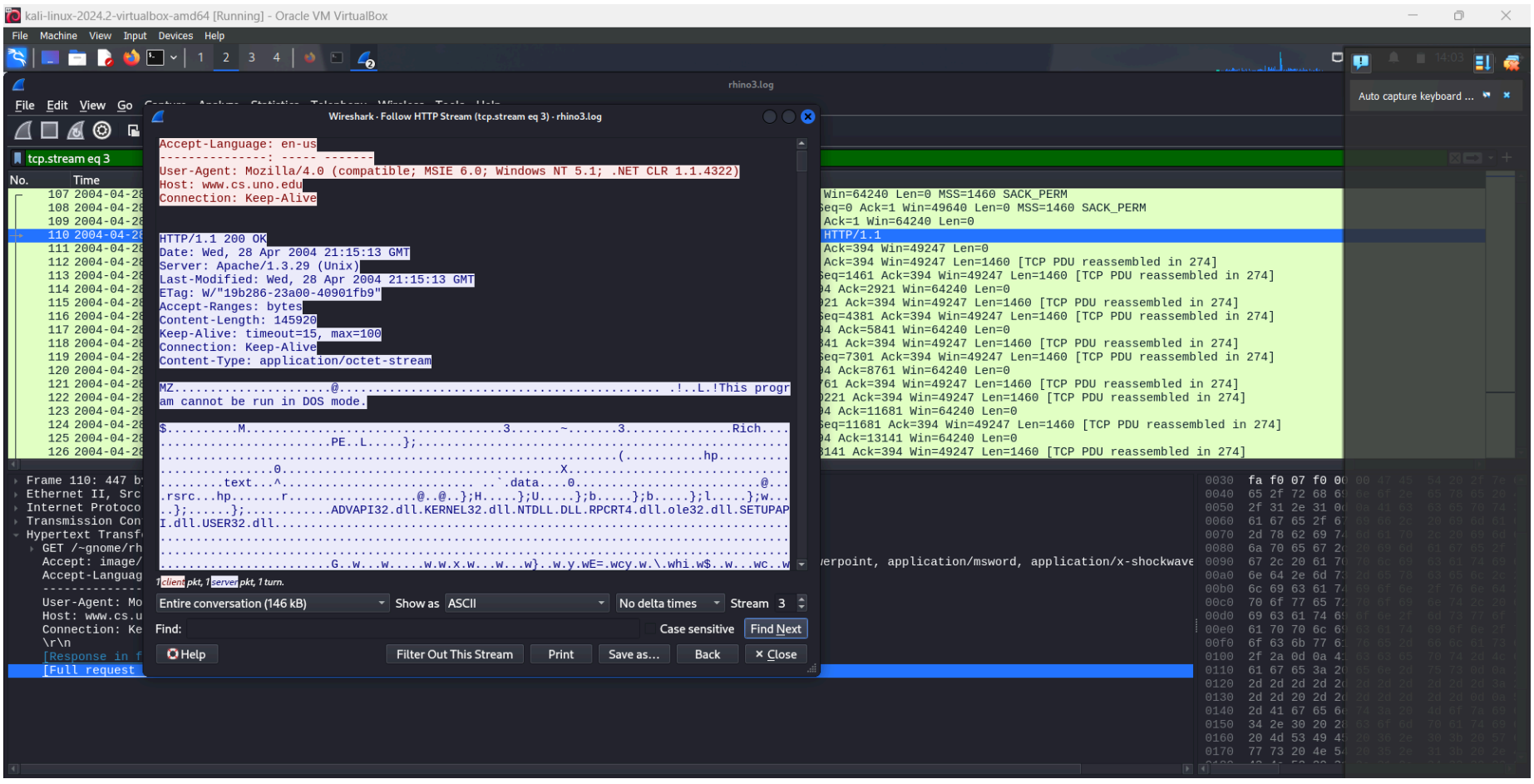
rhino3.log

This is the last and final log file to examine and which has a ".exe" file when running with wine it returns a hash and further

checking the hash we can get the summary below.







```
┌──(root㉿kali)-[/home/…/Rhino_CaseStudy/rhino/Logs_Tracing/Http_Traffic]
└─# ls
rhino2.log  rhino3.log  rhino7.gif  rhino8.exe

┌──(root㉿kali)-[/home/…/Rhino_CaseStudy/rhino/Logs_Tracing/Http_Traffic]
└─# ls - l rhino8.exe
ls: cannot access '-': No such file or directory
ls: cannot access 'l': No such file or directory
rhino8.exe

┌──(root㉿kali)-[/home/…/Rhino_CaseStudy/rhino/Logs_Tracing/Http_Traffic]
└─# ls -l rhino8.exe
-rw-r--r-- 1 root root 145920 Apr  6 14:11 rhino8.exe

┌──(root㉿kali)-[/home/…/Rhino_CaseStudy/rhino/Logs_Tracing/Http_Traffic]
└─# wine rhino8.exe

Microsoft DiskPart version 1.0
Copyright (C) 1999-2001 Microsoft Corporation.
On computer: KALI
0024:err:ole:com_get_class_object class {4fb6bb00-3347-11d0-b40a-00aa005ff586} not registered
0024:err:ole:create_server class {4fb6bb00-3347-11d0-b40a-00aa005ff586} not registered
0024:err:ole:com_get_class_object no class object {4fb6bb00-3347-11d0-b40a-00aa005ff586} could be created for context 0x1
5

The disk management services could not complete the operation.

┌──(root㉿kali)-[/home/…/Rhino_CaseStudy/rhino/Logs_Tracing/Http_Traffic]
└─#
```

```
┌──(root㉿kali)-[/home/…/Rhino_CaseStudy/rhino/Logs_Tracing/Http_Traffic]
└─# md5sum rhino8.exe
d62d9989535c4c8db14e50b58c9f25a0  rhino8.exe

┌──(root㉿kali)-[/home/…/Rhino_CaseStudy/rhino/Logs_Tracing/Http_Traffic]
└─#
```

d62d9989535c4c8db14e50b58c9f25a0

Summary

Architecture	`IMAGE_FILE_MACHINE_I386`
Subsystem	`IMAGE_SUBSYSTEM_WINDOWS_CUI`
Compilation Date	2001-Aug-17 20:59:36
Detected languages	English - United States
Debug artifacts	.pdb
CompanyName	Microsoft Corporation
FileDescription	Diskpart Application
FileVersion	`1, 0, 3, 1`
InternalName	diskpart
LegalCopyright	Copyright © 2000
OriginalFilename	diskpart.rc
ProductName	Microsoft Corporation Diskpart Application
ProductVersion	`1, 0, 3, 1`

Final Conclusion

What we got ?
Recovered photos : 8 jpg, 3 gif

1.Who gave the accused a telnet/ftp account?
Jeremy (from diary)
2.What's the username/password for the account?
gnome / gnome123
3.What relevant file transfers appear in the network traces?
rhino1.jpg, rhino3.jpg in rhino.log trace
rhino2.jpg in contraband.zip file from rhino.log trace
rhino4.jpg, rhino5.gif in rhino2.log
An executable "rhino.exe" in rhino3.log
4.What happened to the hard drive in the computer? Where is it now?
Suspect tossed it into the Mississippi River (from diary)
5.What happened to the USB key?
Suspect reformatted it—possibly at Radio Shack--hoping not to
overwrite the "good" stuff. Source: diary.
6.What is recoverable from the dd image of the USB key?

rhino6.jpg in alligator2.jpg
rhino7.jpg in alligator3.jpg
rhino2, rhino8.gif, rhino9.gif, rhino10.bmp
Word document “diary.doc” contains some answers to the questions

7.Is there any evidence that connects the USB key and the network traces? If so, what?

At least one thing:
rhino2.jpg carved from USBKEY is same as rhino2.jpg in zip file from
network trace.