



FORENSICS LAB SERIES

Lab 10: Network Forensics

Material in this Lab Aligns to the Following Certification Domains/Objectives		
GIAC Certified Forensics Examiner (GCFE) Domains	Certified Cyber Forensics Professional (CCFP) Objectives	Computer Hacking Forensic Investigator (CHFI) Objectives
7: User Communication Analysis	4: Digital Forensics	16: Network Forensics, Investigating Logs and Investigating Network Traffic

Document Version: 2016-08-17

Copyright © 2016 Network Development Group, Inc.
www.netdevgroup.com

NETLAB Academy Edition, NETLAB Professional Edition and NETLAB+ are registered trademarks of Network Development Group, Inc.

Contents

Introduction	3
Objective	3
Pod Topology	4
Lab Settings	5
1 Capturing and Analyzing Traffic with Tcpdump.....	6
2 Analyzing Traffic with Wireshark	11

Introduction

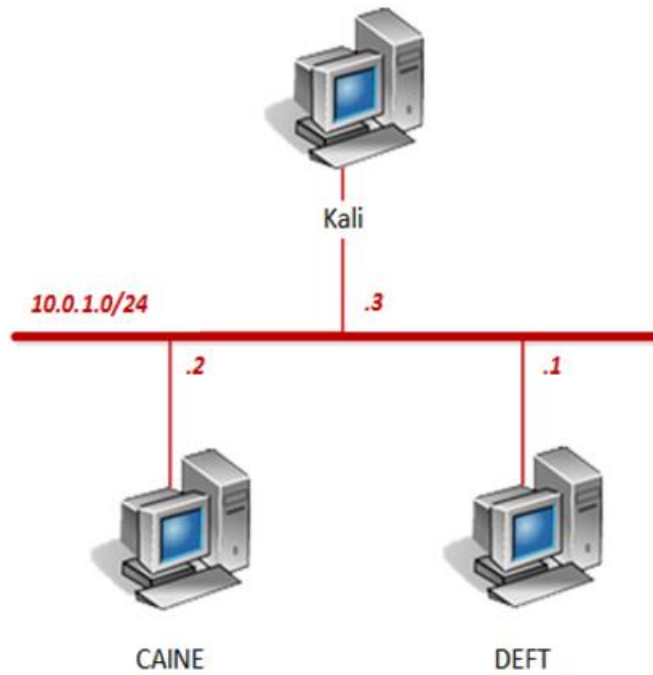
This lab will introduce how to capture packets and interpret them when performing network forensics. We will also examine how to use a graphical network analyzer to interpret the results.

Objective

In this lab, you will be conducting forensic practices using various tools. You will be performing the following tasks:

1. Capturing and Analyzing Traffic with Tcpdump
2. Analyzing Traffic with Wireshark

Pod Topology



Lab Settings

The information in the table below will be needed in order to complete the lab. The task sections below provide details on the use of this information.

Virtual Machine	IP Address	Account (if needed)	Password (if needed)
DEFT	10.0.1.1	deft	password
CAINE	10.0.1.2	caine	
Kali	10.0.1.3	root	toor

1 Capturing and Analyzing Traffic with Tcpcdump

1. Click on the **CAINE** graphic on the *topology page* to open the VM.
2. Open a new terminal by clicking on the **MATE Terminal** icon located in the bottom tool pane.



3. Observe the manual for the *tcpcdump* application. Enter the command below followed by pressing the **Enter** key.

```
man tcpcdump
```

```
TCPDUMP(8)                      System Manager's Manual                      TCPDUMP(8)

NAME
    tcpcdump - dump traffic on a network

SYNOPSIS
    tcpcdump [ -AbDeFhHIJKLlNOpqRStuUvX ] [ -B buffer_size ] [ -c count ]
    [ -C file_size ] [ -G rotate_seconds ] [ -F file ]
    [ -i interface ] [ -j tstamp_type ] [ -m module ] [ -H secret ]
    [ -P in|out|inout ]
    [ -r file ] [ -V file ] [ -s snaplen ] [ -T type ] [ -w file ]
    [ -W filecount ]
    [ -E spi@ipaddr algo:secret,... ]
    [ -y datalinktype ] [ -z postrotate-command ] [ -Z user ]
    [ expression ]

DESCRIPTION
    Tcpcdump prints out a description of the contents of packets on a net-
    work interface that match the boolean expression. It can also be run
    with the -w flag, which causes it to save the packet data to a file for
    later analysis, and/or with the -r flag, which causes it to read from a
    saved packet file rather than to read packets from a network interface
    (please note tcpcdump is protected via an enforcing apparmor(7) profile
    in Ubuntu which limits the files tcpcdump may access). It can also be
    run with the -V flag, which causes it to read a list of saved packet
    files. In all cases, only packets that match expression will be pro-
    cessed by tcpcdump.

Manual page tcpcdump(8) line 1 (press h for help or q to quit)
```

With the *man* command, use the **Enter** key to skip to the next line item or use the **spacebar** to skip by page. When finished, press the **q** character to quit.

4. *Tcpdump* can perform network captures as well as filtering specific traffic. Enter the command below to try a simple capture.

```
sudo tcpdump -i eth0 -s0 -v
```

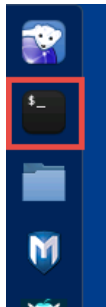
```
caine@Caine01:~$ sudo tcpdump -i eth0 -s0 -v
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 65535 byte
S
```

Command breakdown:

-i means interface
-s0 means capture all bytes within the packet
-v mean results should be verbose

If prompted for a password, type `password` and press the **Enter** key.

5. Change focus to the topology page and click on the **Kali** graphic to open the VM.
6. Login using `root` as the *username* and `toor` as the *password*.
7. Open a new terminal by clicking on the **Terminal** icon located in the left tool pane.



8. Using the terminal, enter the command below to continuously ping the CAINE system.

```
ping 10.0.1.2
```

```
root@Kali2:~# ping 10.0.1.2
PING 10.0.1.2 (10.0.1.2) 56(84) bytes of data.
64 bytes from 10.0.1.2: icmp_seq=1 ttl=64 time=2.60 ms
64 bytes from 10.0.1.2: icmp_seq=2 ttl=64 time=0.140 ms
64 bytes from 10.0.1.2: icmp_seq=3 ttl=64 time=0.190 ms
```

Leave this running in the background.



9. Change focus to the **CAINE** PC viewer.
10. Notice on the terminal window, *tcpdump* is reportedly capturing *ICMP* traffic from the *Kali* system. Press **CTRL+C** to stop the capture.

```
18:34:18.282094 IP (tos 0x0, ttl 64, id 14378, offset 0, flags [none], proto ICMP (1), length 84)
    10.0.1.2 > 10.0.1.3: ICMP echo reply, id 1796, seq 96, length 64
18:34:19.282077 IP (tos 0x0, ttl 64, id 1266, offset 0, flags [DF], proto ICMP (1), length 84)
    10.0.1.3 > 10.0.1.2: ICMP echo request, id 1796, seq 97, length 64
18:34:19.282109 IP (tos 0x0, ttl 64, id 14603, offset 0, flags [none], proto ICMP (1), length 84)
    10.0.1.2 > 10.0.1.3: ICMP echo reply, id 1796, seq 97, length 64
18:34:20.282063 IP (tos 0x0, ttl 64, id 1381, offset 0, flags [DF], proto ICMP (1), length 84)
    10.0.1.3 > 10.0.1.2: ICMP echo request, id 1796, seq 98, length 64
18:34:20.282091 IP (tos 0x0, ttl 64, id 14843, offset 0, flags [none], proto ICMP (1), length 84)
    10.0.1.2 > 10.0.1.3: ICMP echo reply, id 1796, seq 98, length 64
^C
272 packets captured
274 packets received by filter
0 packets dropped by kernel
caine@Caine01:~$
```

11. Enter the command below to dump captured packets to a **.pcap** file type.

```
sudo tcpdump -i eth0 -s0 -v > dump1.pcap
```

If prompted for a password, type **password** and press the **Enter** key.

12. Wait for about 1-2 minutes of capturing traffic and press **CTRL+C** to stop the *tcpdump* capture.

```
caine@Caine01:~$ sudo tcpdump -i eth0 -s0 -v > dump1.pcap
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 65535 bytes
^C722 packets captured
724 packets received by filter
0 packets dropped by kernel
caine@Caine01:~$
```

Notice 724 packets have been captured from this example.



13. Analyze only the *ICMP* traffic from the *dump1.pcap* file by using the *grep* command. Enter the command below.

```
cat dump1.pcap | grep ICMP | less
```

```
18:45:27.282071 IP (tos 0x0, ttl 64, id 17703, offset 0, flags [DF], proto ICMP
(1), length 84)
    10.0.1.3 > 10.0.1.2: ICMP echo request, id 1796, seq 765, length 64
18:45:27.282100 IP (tos 0x0, ttl 64, id 29726, offset 0, flags [none], proto ICM
P (1), length 84)
    10.0.1.2 > 10.0.1.3: ICMP echo reply, id 1796, seq 765, length 64
18:45:28.282064 IP (tos 0x0, ttl 64, id 17859, offset 0, flags [DF], proto ICMP
(1), length 84)
    10.0.1.3 > 10.0.1.2: ICMP echo request, id 1796, seq 766, length 64
18:45:28.282094 IP (tos 0x0, ttl 64, id 29752, offset 0, flags [none], proto ICM
P (1), length 84)
    10.0.1.2 > 10.0.1.3: ICMP echo reply, id 1796, seq 766, length 64
18:45:29.282065 IP (tos 0x0, ttl 64, id 17962, offset 0, flags [DF], proto ICMP
(1), length 84)
    10.0.1.3 > 10.0.1.2: ICMP echo request, id 1796, seq 767, length 64
18:45:29.282091 IP (tos 0x0, ttl 64, id 29883, offset 0, flags [none], proto ICM
P (1), length 84)
    10.0.1.2 > 10.0.1.3: ICMP echo reply, id 1796, seq 767, length 64
18:45:30.282052 IP (tos 0x0, ttl 64, id 18122, offset 0, flags [DF], proto ICMP
(1), length 84)
    10.0.1.3 > 10.0.1.2: ICMP echo request, id 1796, seq 768, length 64
18:45:30.282080 IP (tos 0x0, ttl 64, id 29988, offset 0, flags [none], proto ICM
P (1), length 84)
    10.0.1.2 > 10.0.1.3: ICMP echo reply, id 1796, seq 768, length 64
18:45:31.282093 IP (tos 0x0, ttl 64, id 18338, offset 0, flags [DF], proto ICMP
(1), length 84)
    10.0.1.3 > 10.0.1.2: ICMP echo request, id 1796, seq 769, length 64
:|
```

With the *less* command, use the **Enter** key to skip to the next line item or use the **spacebar** to skip by page. When finished, press the **q** character to quit.

14. Initiate a new capture with *tcpdump* that can be opened in a graphical network analyzer. Enter the command below.

```
sudo tcpdump -i eth0 -s 65535 -w dump2.pcap
```

```
caine@Caine01:~$ sudo tcpdump -i eth0 -s 65535 -w dump2.pcap
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 65535 byte
S
|
```

Command breakdown:

-s means capture 65535 bytes
-w means write to a file

If prompted for a password, type **password** and press the **Enter** key.

15. Wait for about 1-2 minutes of capturing traffic and press **CTRL+C** to stop the *tcpdump* capture.

```
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 65535 bytes
^C240 packets captured
240 packets received by filter
0 packets dropped by kernel
caine@Caine01:~$
```

Notice 240 packets have been captured from this example.

16. Leave the terminal open to continue with the next task.

2 Analyzing Traffic with Wireshark

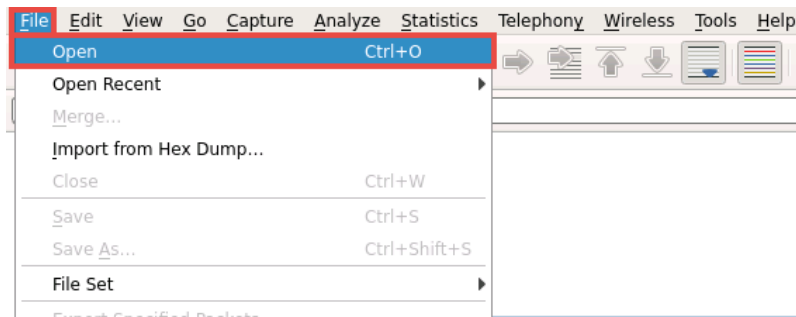
1. Using the terminal, enter the command below to launch a GUI based network analyzer named *Wireshark*.

```
sudo wireshark
```

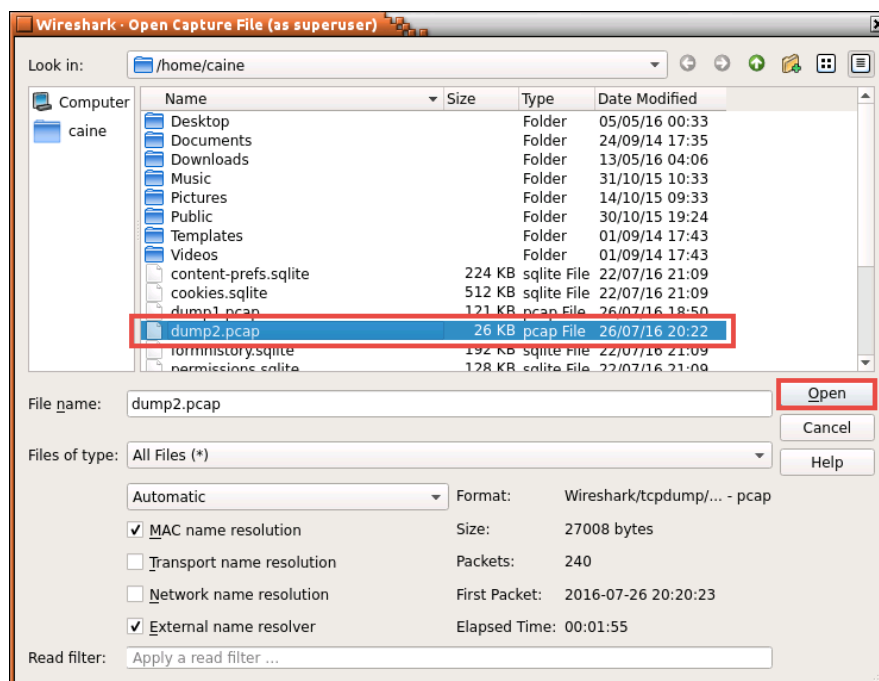
```
caine@Caine01:~$ sudo wireshark
```

If prompted for a password, type `password` and press the **Enter** key.

2. Using Wireshark, click on **File** and select **Open**.



3. In the *Open Capture File* window, navigate to the `/home/caine/` directory and select **dump2.pcap**. Click **Open**.



4. Notice the *ICMP* traffic data fill the middle pane of the *Wireshark* application. Change focus to **Kali** system to generate some additional traffic.

5. Focusing on the terminal, press **CTRL+C** to stop the pings from occurring.
6. Using the terminal, enter the command below to start the *Apache* web service on the *Kali* system.

```
service apache2 start
```

```
root@Kali2:~# service apache2 start
root@Kali2:~#
```

If *Apache* returns with an error, enter the following command and continue to the next step: **service apache2 restart**

7. Verify the status of the *Apache* service by entering the command below.

```
root@Kali2:~# service apache2 status
● apache2.service - LSB: Apache2 web server
   Loaded: loaded (/etc/init.d/apache2)
   Active: active (running) since Wed 2016-07-27 10:17:10 CDT; 21min ago
   Process: 1683 ExecStart=/etc/init.d/apache2 start (code=exited, status=0/SUCCESS)
   CGroup: /system.slice/apache2.service
           └─1704 /usr/sbin/apache2 -k start
             └─1708 /usr/sbin/apache2 -k start
               └─1709 /usr/sbin/apache2 -k start
                 └─1710 /usr/sbin/apache2 -k start
                   └─1711 /usr/sbin/apache2 -k start
                     └─1712 /usr/sbin/apache2 -k start
                       └─1713 /usr/sbin/apache2 -k start

Jul 27 10:17:10 Kali2 apache2[1683]: Starting web server: apache2.
root@Kali2:~#
```

8. Start the *FTP* service on the *Kali* system. Enter the command below.

```
service vsftpd start
```

```
root@Kali2:~# service vsftpd start
root@Kali2:~#
```

9. Enter the command below to verify the status of the *FTP* service.

```
service vsftpd status
```

```
root@Kali2:~# service vsftpd status
● vsftpd.service - vsftpd FTP server
   Loaded: loaded (/lib/systemd/system/vsftpd.service; disabled)
   Active: active (running) since Wed 2016-07-27 10:41:22 CDT; 27s ago
   Process: 1790 ExecStartPre=/bin/mkdir -p /var/run/vsftpd/empty (code=exited, status=0/SUCCESS)
   Main PID: 1792 (vsftpd)
   CGroup: /system.slice/vsftpd.service
           └─1792 /usr/sbin/vsftpd /etc/vsftpd.conf

root@Kali2:~#
```

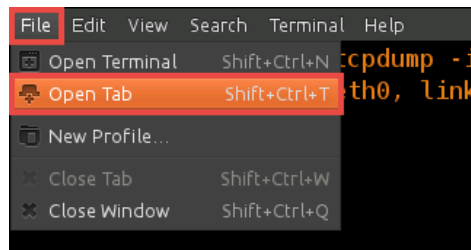
10. Change focus to the **CAINE** system.
11. Close the **Wireshark** application.
12. Using the terminal, enter the command below to initiate a new *tcpdump* capture.

```
sudo tcpdump -i eth0 -s 65535 -w dump3.pcap
```

```
caine@Caine01:~$ sudo tcpdump -i eth0 -s 65535 -w dump3.pcap
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 65535 bytes
```

If prompted for a password, type **password** and press the **Enter** key.

13. In the terminal window, click on **File** and select **Open Tab**.



14. In the new tab of the terminal window, enter the command below to initiate an ftp connection to the Kali FTP service.

```
ftp 10.0.1.3
```

```
caine@Caine01:~$ ftp 10.0.1.3
Connected to 10.0.1.3.
220 Welcome to Kali FTP service.
Name (10.0.1.3:caine):
```

15. When prompted with a name field, type **caine** and press the **Enter** key.

```
Name (10.0.1.3:caine): caine
331 Please specify the password.
Password:
```

16. When prompted for a password, type **password** and press **Enter**.

```
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

17. Once the **ftp>** prompt appears, enter the command below to list the files and directories.

```
ls
```

```
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
drwxr-xr-x  2 1000      1001      4096 May 20 10:31 Desktop
drwxr-xr-x  2 1000      1001      4096 May 20 10:56 Documents
drwxr-xr-x  2 1000      1001      4096 May 20 10:31 Downloads
drwxr-xr-x  2 1000      1001      4096 May 20 10:31 Music
drwxr-xr-x  2 1000      1001      4096 May 20 10:31 Pictures
drwxr-xr-x  2 1000      1001      4096 May 20 10:31 Public
drwxr-xr-x  2 1000      1001      4096 May 20 10:31 Templates
drwxr-xr-x  2 1000      1001      4096 May 20 10:31 Videos
226 Directory send OK.
ftp>
```

18. Navigate to the Documents/ directory by entering the command below.

```
cd Documents
```

```
ftp> cd Documents
250 Directory successfully changed.
ftp>
```

19. List the files and directories again. Enter the command below.

```
ls
```

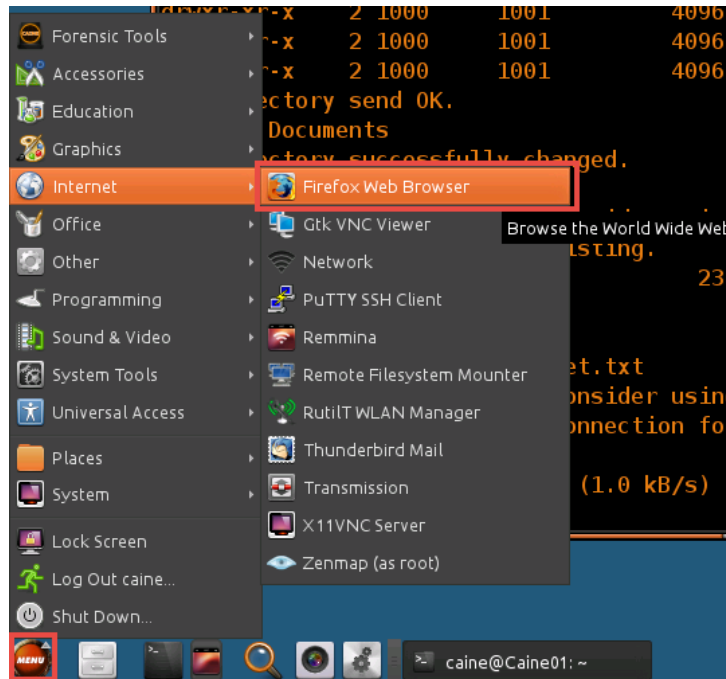
```
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
-rw-r--r--  1 0        0        23 May 20 10:56 secret.txt
226 Directory send OK.
ftp>
```

20. Notice the *secret.txt* file. Download this file locally by entering the command below.

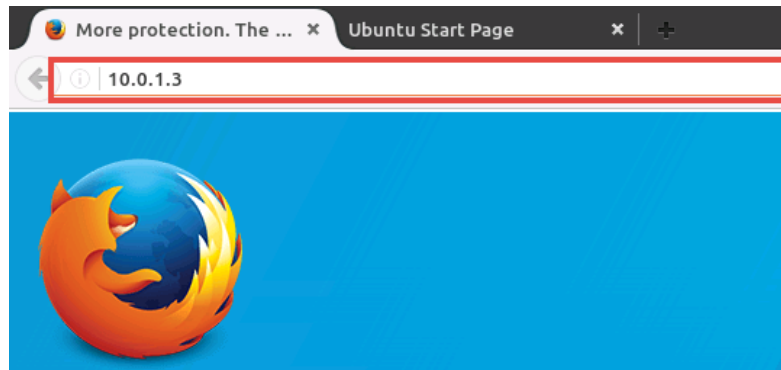
```
get secret.txt
```

```
ftp> get secret.txt
local: secret.txt remote: secret.txt
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for secret.txt (23 bytes).
226 Transfer complete.
23 bytes received in 0.02 secs (1.0 kB/s)
ftp>
```

21. Open the *Firefox* web browser by navigating to **Menu > Internet > Firefox Web Browser**.



22. Using *Firefox*, type `10.0.1.3` into the address field and press **Enter**.



23. Notice the *Apache2 Debian Default Page* appears. Change focus to the terminal window in *CAINE* and click on the **first tab**.
24. Press **CTRL+C** to stop the *tcpdump* capture.

```
caine@Caine01:~$ sudo tcpdump -i eth0 -s 65535 -w dump3.pcap
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 65535 bytes
^C1472 packets captured
1472 packets received by filter
0 packets dropped by kernel
caine@Caine01:~$
```

If prompted for a password, type `password` and press the **Enter** key.

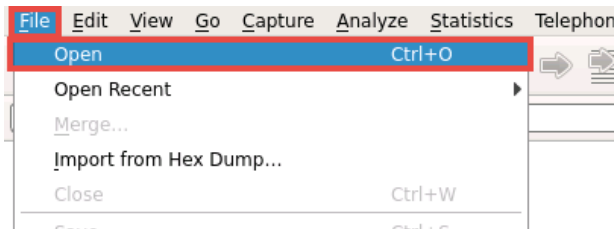
25. Using the terminal, enter the command below to open **Wireshark**.

```
sudo wireshark
```

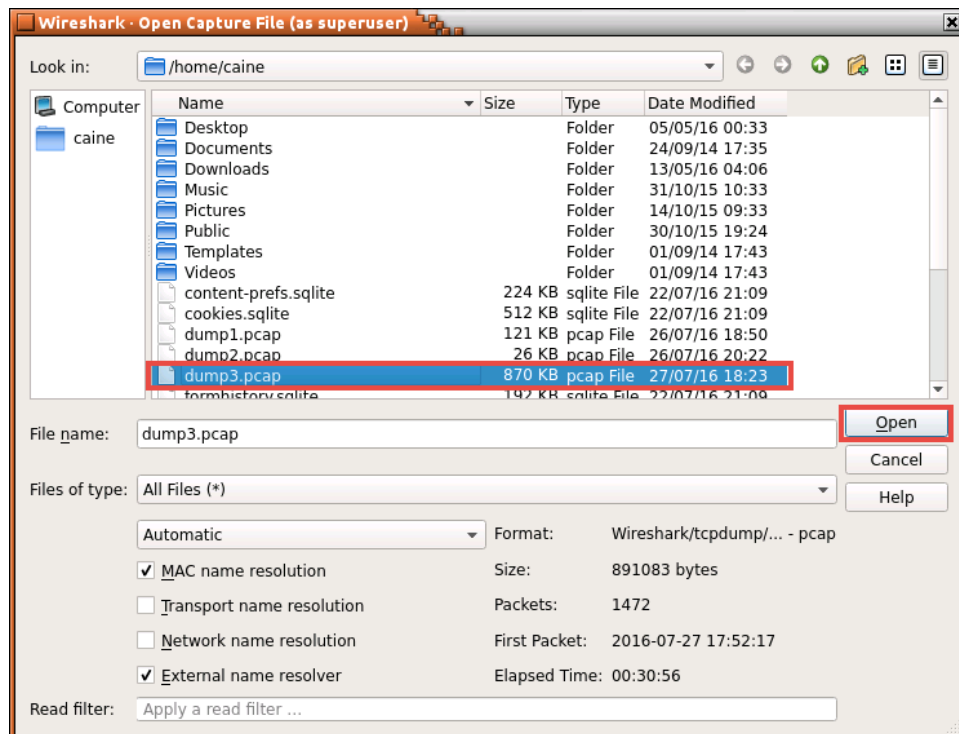
```
caine@Caine01:~$ sudo wireshark
```

If prompted for a password, type password and press the Enter key.

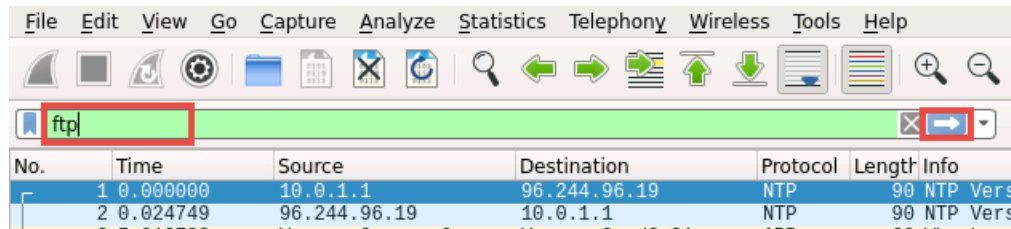
26. Using the *Wireshark* application, click **File** and select **Open**.



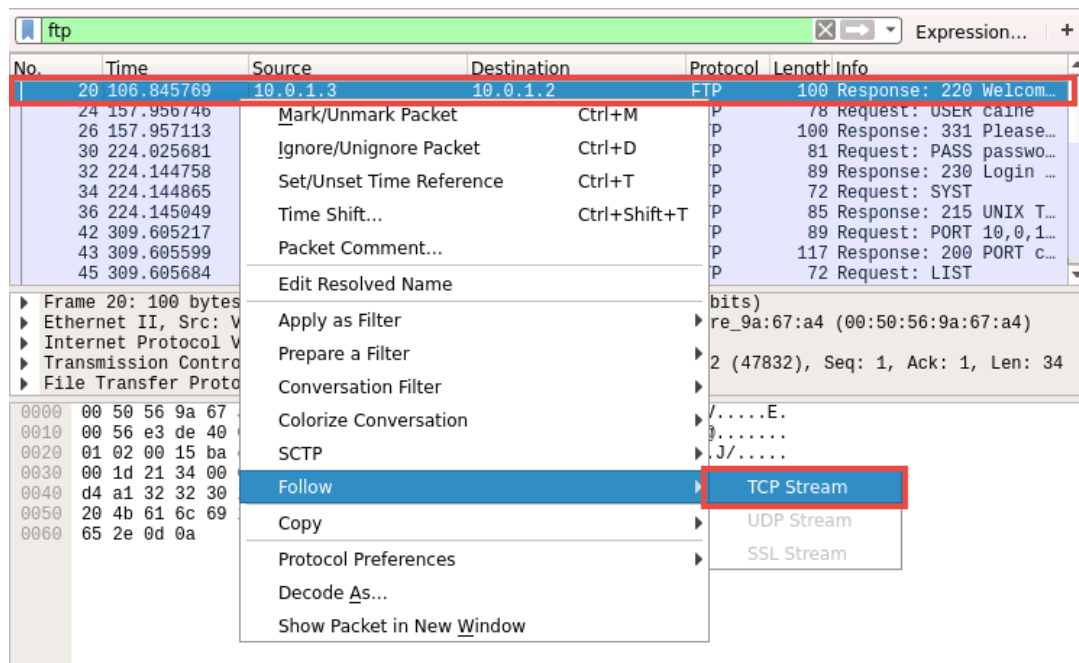
27. In the *Open Capture File* window, navigate to the **/home/caine/** directory and select **dump3.pcap**. Click **Open**.



28. Notice the traffic data populating the middle pane. Filter the traffic for just *FTP* traffic by typing **ftp** into the filter field and clicking on the **blue arrow** to apply.

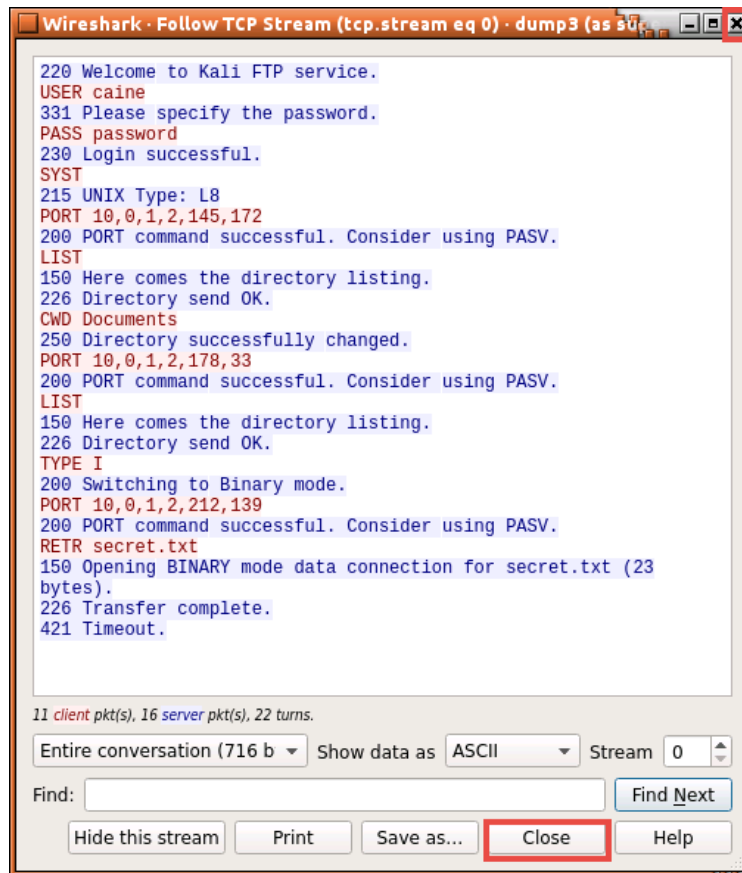


29. Only traffic with the *FTP* protocol should appear in the middle pane. Analyze the stream of data that happened by right-clicking on the **first FTP frame** and selecting **Follow > TCP Stream**.





30. Notice in a new window, the entire *FTP* conversation appears in clear text. The login and password can be seen, along with which file was downloaded and its corresponding filename. Close the window.



```

220 Welcome to Kali FTP service.
USER caine
331 Please specify the password.
PASS password
230 Login successful.
SYST
215 UNIX Type: L8
PORT 10,0,1,2,145,172
200 PORT command successful. Consider using PASV.
LIST
150 Here comes the directory listing.
226 Directory send OK.
CWD Documents
250 Directory successfully changed.
PORT 10,0,1,2,178,33
200 PORT command successful. Consider using PASV.
LIST
150 Here comes the directory listing.
226 Directory send OK.
TYPE I
200 Switching to Binary mode.
PORT 10,0,1,2,212,139
200 PORT command successful. Consider using PASV.
RETR secret.txt
150 Opening BINARY mode data connection for secret.txt (23
bytes).
226 Transfer complete.
421 Timeout.
  
```

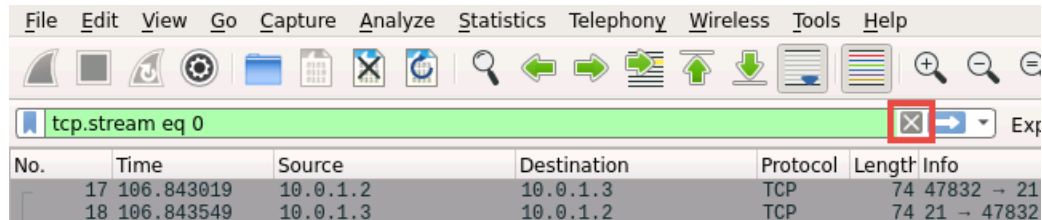
11 client pkt(s), 16 server pkt(s), 22 turns.

Entire conversation (716 b) Show data as ASCII Stream 0

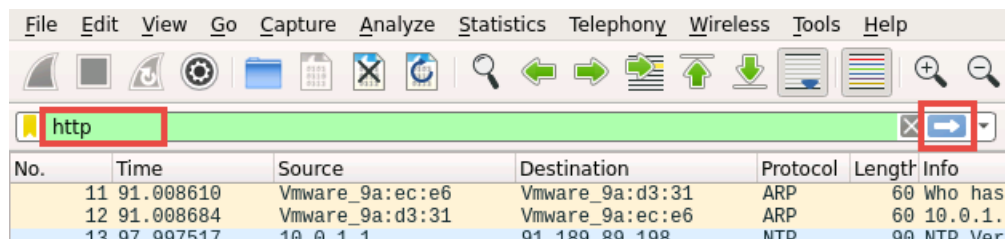
Find: Find Next

Hide this stream Print Save as... Close Help

31. Clear the filter field by clicking on the “x” symbol next to the blue arrow.



32. Type **http** into the filter field followed by clicking on the **blue arrow** to apply.





33. Notice that only traffic with the *HTTP* protocol should appear in the middle pane. Briefly analyze the data presented.

http						Expression...	+
No.	Time	Source	Destination	Protocol	Length	Info	
201	1608.266714	10.0.1.2	204.85.32.40	HTTP	155	GET /mgetmetar.php?c...	
203	1608.314839	204.85.32.40	10.0.1.2	HTTP	7681	HTTP/1.1 200 OK (te...	
261	1715.910392	10.0.1.2	72.21.91.29	OCSP	505	Request	
263	1715.921221	72.21.91.29	10.0.1.2	OCSP	854	Response	
265	1715.935685	10.0.1.2	91.189.89.88	HTTP	472	GET /14.04/Google/?s...	
287	1716.033160	91.189.89.88	10.0.1.2	HTTP	407	HTTP/1.0 304 Not Mod...	
291	1716.110291	10.0.1.2	72.21.91.29	OCSP	505	Request	
294	1716.120795	72.21.91.29	10.0.1.2	OCSP	854	Response	
305	1716.243738	10.0.1.2	91.189.89.88	HTTP	433	GET /12.04/sprite.pn...	
311	1716.338375	91.189.89.88	10.0.1.2	HTTP	408	HTTP/1.0 304 Not Mod...	
▶ Frame 201: 155 bytes on wire (1240 bits), 155 bytes captured (1240 bits) ▶ Ethernet II, Src: Vmware_9a:67:a4 (00:50:56:9a:67:a4), Dst: Vmware_9a:d3:31 (00:50:56:9a:d3:31) ▶ Internet Protocol Version 4, Src: 10.0.1.2, Dst: 204.85.32.40 ▶ Transmission Control Protocol, Src Port: 34864 (34864), Dst Port: 80 (80), Seq: 1, Ack: 1, Len: 89 ▶ Hypertext Transfer Protocol							
0000	00 50 56 9a d3 31 00 50	56 9a 67 a4 08 00 45 00	.PV..1.P V.g...E.				
0010	00 8d 86 c8 40 00 40 06	bc 23 0a 00 01 02 cc 55	...@.@. .#....U				
0020	20 28 88 30 00 50 ed d1	08 1b 44 11 86 7d 80 18	(.0.P.. ..D..)..				
0030	00 e5 f7 fe 00 00 01 01	08 0a 00 1a 8e dd 32 8c2.				
0040	7b 85 47 45 54 20 2f 6d	67 65 74 6d 65 74 61 72	{.GET /m getmetar				
0050	2e 70 68 70 3f 63 63 63	63 3d 4b 4d 44 57 20 48	.php?ccc c=KMDW H				
0060	54 54 50 2f 31 2e 31 0d	0a 48 6f 73 74 3a 20 77	TTP/1.1. .Host: w				
0070	65 61 74 68 65 72 2e 6e	6f 61 61 2e 67 6f 76 0d	eather.n oaa.gov.				
0080	0a 43 6f 6e 6e 65 63 74	69 6f 6e 3a 20 4b 65 65	.Connect ion: Kee				
0090	70 2d 41 6c 69 76 65 0d	0a 0d 0a	p-Alive. ...				

34. Close all **PC Viewers** and end the reservation to complete the lab.