Intro to Network Traffic Analysis:

Link to challenge: https://academy.hackthebox.com/module/81

(log in required)

Class: Tier 0 | Medium | General

Introduction

Networking Primer - Layers 1-4:

Question: How many layers does the OSI model have?

Answer: 7

Method: Physical, Link, Network, Transport, Session, Presentation, Application

Question: How many layers are there in the TCP/IP model?

Answer: 4

Method: Link, Internet, Transport, Application

Question: True or False: Routers operate at layer 2 of the OSI model?

Answer: False

Method: Routers operate at the later Network – Layer 3 of the OSI model.

Question: What addressing mechanism is used at the Link Layer of the TCP/IP model?

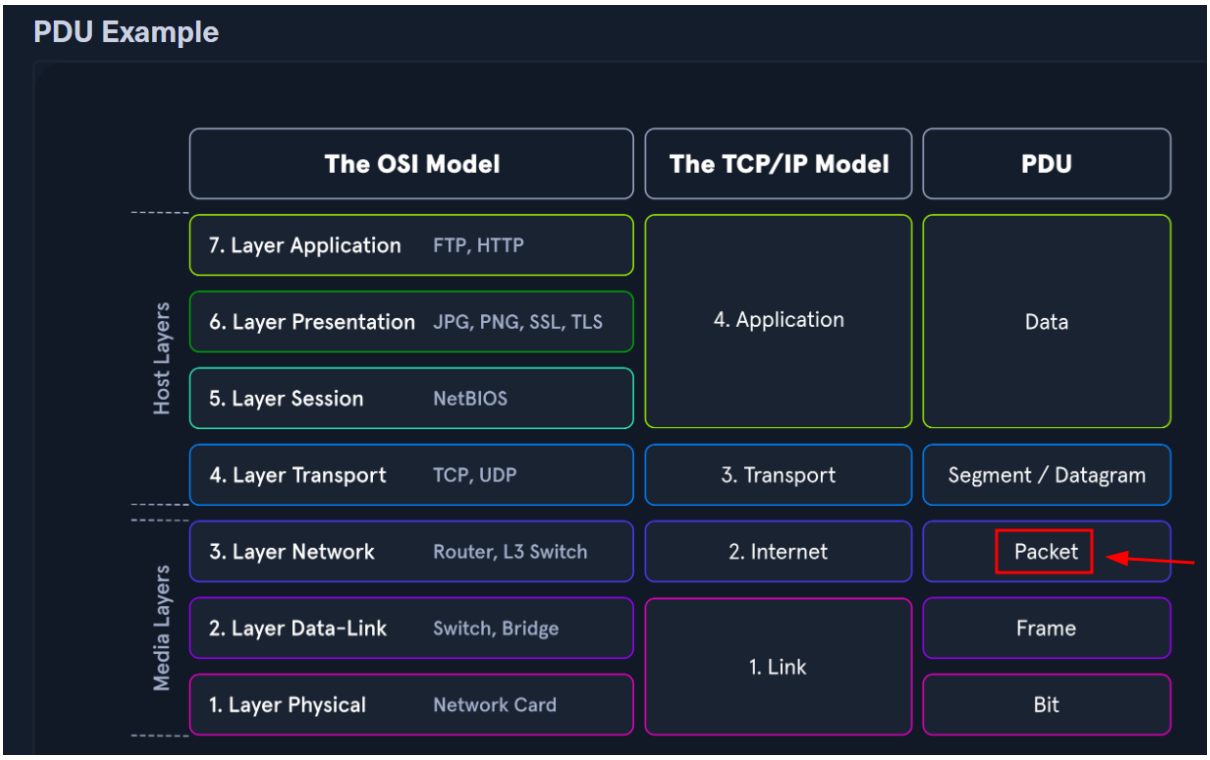
Answer: Mac-Address

Method: 'MAC-addressing is utilized in Layer two (the data-link or link-layer depending on which model you look at) communications between hosts.'

Question: At what layer of the OSI model is a PDU encapsulated into a packet? (the number)

Answer: 3

Method: Network layer:



PDU Example

	The OSI Model		The TCP/IP Model	PDU
Host Layers	7. Layer Application	FTP, HTTP	4. Application	Data
	6. Layer Presentation	JPG, PNG, SSL, TLS		
	5. Layer Session	NetBIOS		
	4. Layer Transport	TCP, UDP	3. Transport	Segment / Datagram
Media Layers	3. Layer Network	Router, L3 Switch	2. Internet	Packet
	2. Layer Data-Link	Switch, Bridge	1. Link	Frame
	1. Layer Physical	Network Card		Bit

Question: What addressing mechanism utilizes a 32-bit address?

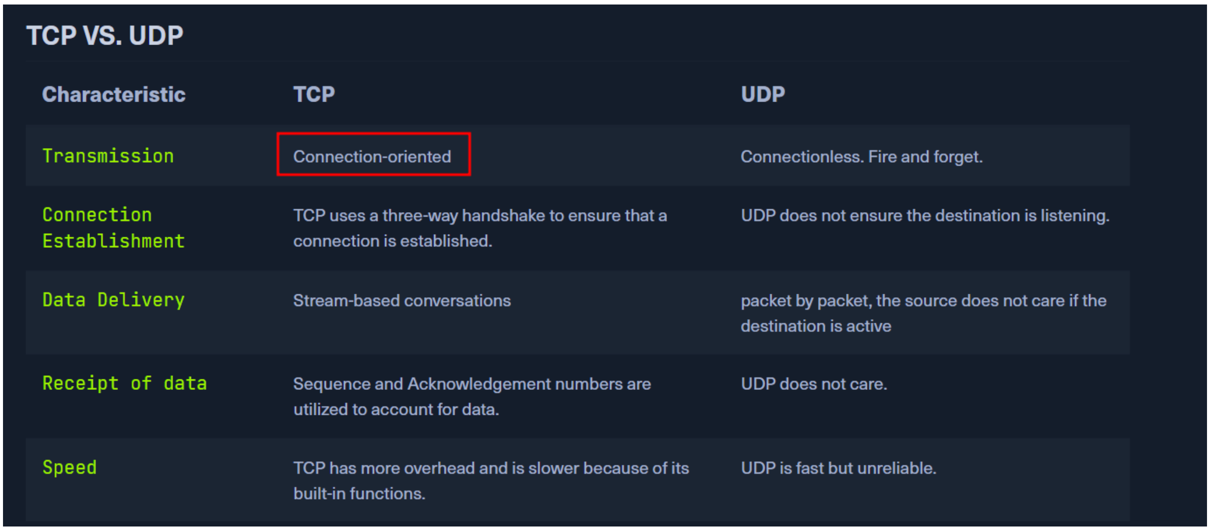
Answer: IPv4

Method: 'An IPv4 address is made up of a 32-bit four octet number represented in decimal format.'

Question: What Transport layer protocol is connection oriented?

Answer: TCP

Method:



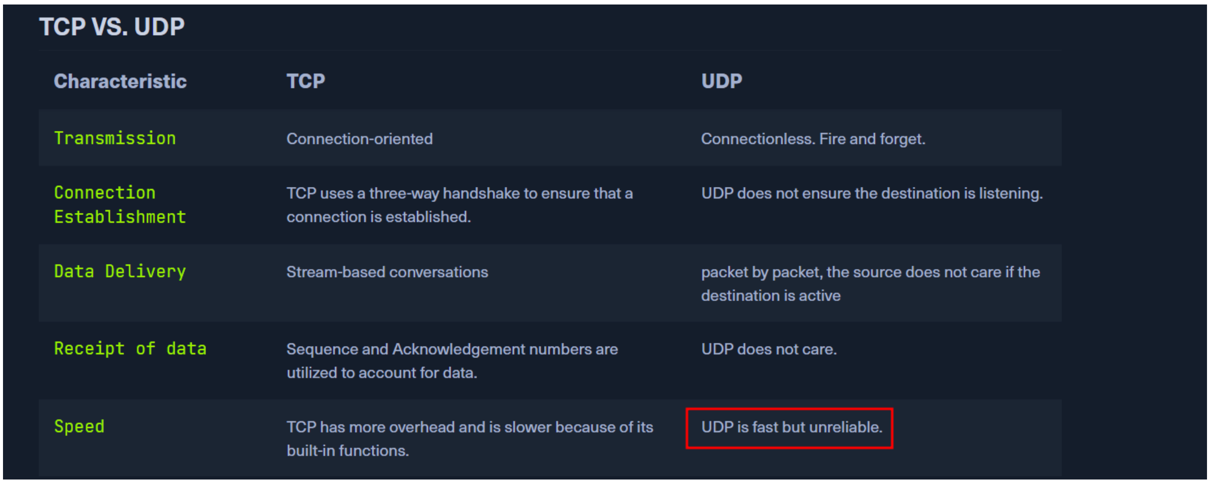
TCP VS. UDP

Characteristic	TCP	UDP
Transmission	Connection-oriented	Connectionless. Fire and forget.
Connection Establishment	TCP uses a three-way handshake to ensure that a connection is established.	UDP does not ensure the destination is listening.
Data Delivery	Stream-based conversations	packet by packet, the source does not care if the destination is active
Receipt of data	Sequence and Acknowledgement numbers are utilized to account for data.	UDP does not care.
Speed	TCP has more overhead and is slower because of its built-in functions.	UDP is fast but unreliable.

Question: What Transport Layer protocol is considered unreliable?

Answer: UDP

Method:



TCP VS. UDP

Characteristic	TCP	UDP
Transmission	Connection-oriented	Connectionless. Fire and forget.
Connection Establishment	TCP uses a three-way handshake to ensure that a connection is established.	UDP does not ensure the destination is listening.
Data Delivery	Stream-based conversations	packet by packet, the source does not care if the destination is active
Receipt of data	Sequence and Acknowledgement numbers are utilized to account for data.	UDP does not care.
Speed	TCP has more overhead and is slower because of its built-in functions.	UDP is fast but unreliable.

Question: TCP's three-way handshake consists of 3 packets: 1.Syn, 2.Syn & ACK, 3. _? What is the final packet of the handshake?

Answer: ACK

Method: the steps are 1. SYN, 2. SYN-ACK, 3. ACK.

Networking Primer - Layers 5-7:

Question: What is the default operational mode method used by FTP?

Answer: Active

Method: ‘Active is the default operational method utilized by FTP’

Question: FTP utilizes what two ports for command and data transfer? (separate the two numbers with a space)

Answer: 20 21

Method: ‘FTP uses ports 20 and 21 over TCP.’

Question: Does SMB utilize TCP or UDP as its transport layer protocol?

Answer: TCP

Method: ‘FTP uses ports 20 and 21 over TCP.’

Question: SMB has moved to using what TCP port?

Answer: 445

Method: ‘SMB now supports direct TCP transport over port 445’

Question: Hypertext Transfer Protocol uses what well known TCP port number?

Answer: 80

Method: ‘HTTP utilizes ports 80 or 8000 over TCP during normal operations.’

Question: What HTTP method is used to request information and content from the webserver?

Answer: GET

Method: ‘Get is the most common method used. It requests information and content from the server.’

Question: What web based protocol uses TLS as a security measure?

Answer: HTTPS

Method: 'HTTP Secure (HTTPS) is a modification of the HTTP protocol designed to utilize Transport Layer Security (TLS) or Secure Sockets Layer (SSL) with older applications for data security.'

Question: True or False: when utilizing HTTPS, all data sent across the session will appear as TLS Application data?

Answer: True

Method: 'Once the session is established, all data and methods will be sent through the TLS connection and appear as TLS Application Data as seen in the red box.'

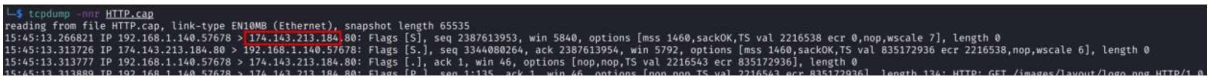
Tcpdump

Tcpdump Fundamentals:

Question: Utilizing the output shown in question-1.png, who is the server in this communication? (IP Address)

Answer: 174.143.213.184

Method: the server runs on port 80:

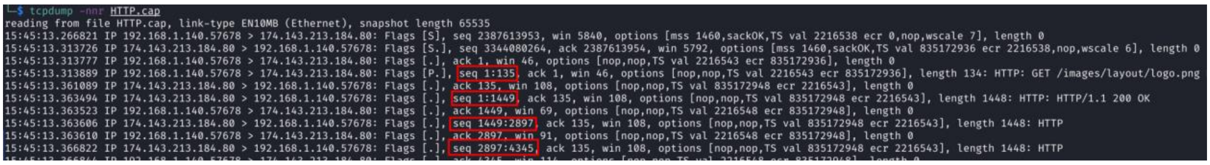


```
└─$ tcpdump -nnr HTTP.cap
reading from file HTTP.cap, link-type EN10MB (Ethernet), snapshot length 65535
15:45:13.266821 IP 192.168.1.140.57678 > 174.143.213.184.80: Flags [S], seq 2387613953, win 5840, options [mss 1460,sackOK,TS val 2216538 ecr 0,nop,wscale 7], length 0
15:45:13.313726 IP 174.143.213.184.80 > 192.168.1.140.57678: Flags [S.], seq 3344080264, ack 2387613954, win 5792, options [mss 1460,sackOK,TS val 835172936 ecr 2216538,nop,wscale 6], length 0
15:45:13.313777 IP 192.168.1.140.57678 > 174.143.213.184.80: Flags [.], ack 1, win 46, options [nop,nop,TS val 2216543 ecr 835172936], length 0
15:45:13.313889 IP 192.168.1.140.57678 > 174.143.213.184.80: Flags [P.], seq 1:135, ack 1, win 46, options [nop,nop,TS val 2216543 ecr 835172936], length 134: HTTP: GET /images/layout/logo.png HTTP/1.0
```

Question: Were absolute or relative sequence numbers used during the capture? (see question-1.zip to answer)

Answer: relative

Method: after connection was established via 3-way handshake – we can see the use of relative sequencing, indicated by the marked low sequence numbers:

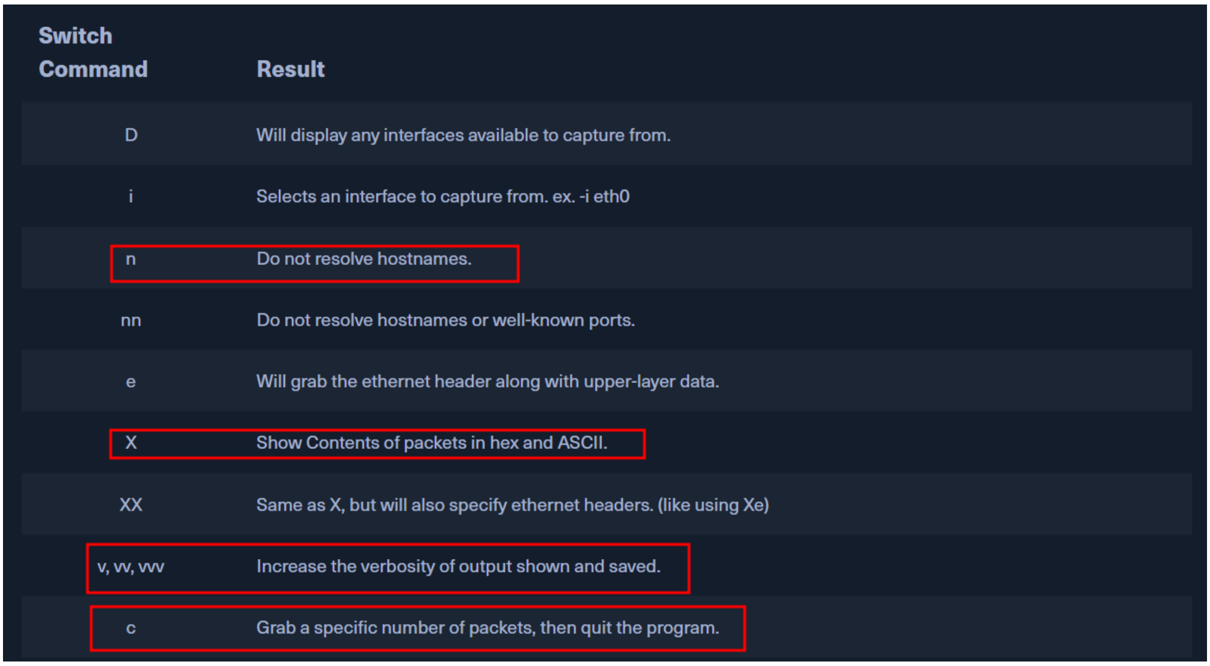


```
└─$ tcpdump -nnr HTTP.cap
reading from file HTTP.cap, link-type EN10MB (Ethernet), snapshot length 65535
15:45:13.266821 IP 192.168.1.140.57678 > 174.143.213.184.80: Flags [S], seq 2387613953, win 5840, options [mss 1460,sackOK,TS val 2216538 ecr 0,nop,wscale 7], length 0
15:45:13.313726 IP 174.143.213.184.80 > 192.168.1.140.57678: Flags [S.], seq 3344080264, ack 2387613954, win 5792, options [mss 1460,sackOK,TS val 835172936 ecr 2216538,nop,wscale 6], length 0
15:45:13.313777 IP 192.168.1.140.57678 > 174.143.213.184.80: Flags [.], ack 1, win 46, options [nop,nop,TS val 2216543 ecr 835172936], length 0
15:45:13.313889 IP 192.168.1.140.57678 > 174.143.213.184.80: Flags [P.], seq 1:135, ack 1, win 46, options [nop,nop,TS val 2216543 ecr 835172936], length 134: HTTP: GET /images/layout/logo.png
15:45:13.361089 IP 174.143.213.184.80 > 192.168.1.140.57678: Flags [.], ack 135, win 108, options [nop,nop,TS val 835172948 ecr 2216543], length 0
15:45:13.363494 IP 174.143.213.184.80 > 192.168.1.140.57678: Flags [.], seq 1:1449, ack 135, win 108, options [nop,nop,TS val 835172948 ecr 2216543], length 1448: HTTP: HTTP/1.1 200 OK
15:45:13.363523 IP 192.168.1.140.57678 > 174.143.213.184.80: Flags [.], ack 1449, win 69, options [nop,nop,TS val 2216548 ecr 835172948], length 0
15:45:13.363606 IP 174.143.213.184.80 > 192.168.1.140.57678: Flags [.], seq 1449:2897, ack 135, win 108, options [nop,nop,TS val 835172948 ecr 2216543], length 1448: HTTP
15:45:13.363610 IP 192.168.1.140.57678 > 174.143.213.184.80: Flags [.], ack 2897, win 91, options [nop,nop,TS val 2216548 ecr 835172948], length 0
15:45:13.366822 IP 174.143.213.184.80 > 192.168.1.140.57678: Flags [.], seq 2897:4345, ack 135, win 108, options [nop,nop,TS val 835172948 ecr 2216543], length 1448: HTTP
15:45:13.366844 IP 192.168.1.140.57678 > 174.143.213.184.80: Flags [.], ack 4345, win 114, options [nop,nop,TS val 2216548 ecr 835172948], length 0
```

Question: If I wish to start a capture without hostname resolution, verbose output, showing contents in ASCII and hex, and grab the first 100 packets; what are the switches used? please answer in the order the switches are asked for in the question.

Answer: -nvXc 100

Method: '-n' for without hostname resolution, '-v' for verbose output, '-X' for ASCII and hex content output, '-c 100' for the first 100 packets:



Switch Command	Result
D	Will display any interfaces available to capture from.
i	Selects an interface to capture from. ex. -i eth0
n	Do not resolve hostnames.
nn	Do not resolve hostnames or well-known ports.
e	Will grab the ethernet header along with upper-layer data.
X	Show Contents of packets in hex and ASCII.
XX	Same as X, but will also specify ethernet headers. (like using Xe)
v, vv, vvv	Increase the verbosity of output shown and saved.
c	Grab a specific number of packets, then quit the program.

Question: Given the capture file at /tmp/capture.pcap, what tcpdump command will enable you to read from the capture and show the output contents in Hex and ASCII? (Please use best practices when using switches)

Answer: sudo tcpdump -Xr /tmp/capture.pcap

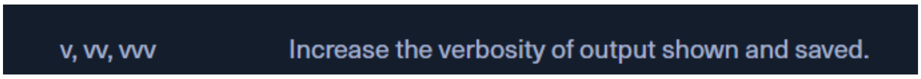
Method: in addition for '-X' switch covered in previous question – the switch of '-r <target-pcap-file>' will read from a file.

We of course use sudo as well.

Question: What TCPDump switch will increase the verbosity of our output? (Include the - with the proper switch)

Answer: -v

Method:



v, vv, vvv	Increase the verbosity of output shown and saved.

Question: What built in terminal help reference can tell us more about TCPDump?

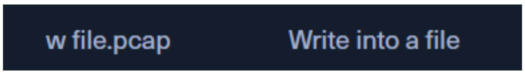
Answer: man

Method: man for manual

Question: What TCPDump switch will let me write my output to a file?

Answer: -w

Method:



w file.pcap	Write into a file

Fundamentals Lab:

Question: What TCPDump switch will allow us to pipe the contents of a pcap file out to another function such as 'grep'?

Answer: -l

Method:

Question: True or False: The filter "port" looks at source and destination traffic.

Answer: True

Method:

Question: If we wished to filter out ICMP traffic from our capture, what filter could we use? (word only, not symbol please.)

Answer: not icmp

Method:

Question: What command will show you where / if TCPDump is installed?

Answer: which tcpdump

Method:

Question: How do you start a capture with TCPDump to capture on eth0?

Answer: tcpdump -i eth0

Method:

Question: What switch will provide more verbosity in your output?

Answer: -v

Method:

Question: What switch will write your capture output to a .pcap file?

Answer: -w

Method:

Question: What switch will read a capture from a .pcap file?

Answer: -r

Method:

Question: What switch will show the contents of a capture in Hex and ASCII?

Answer: -X

Method:

Tcpdump Packet Filtering:

Question: What filter will allow me to see traffic coming from or destined to the host with an ip of 10.10.20.1?

Answer: host 10.10.20.1

Method:

Question: What filter will allow me to capture based on either of two options?

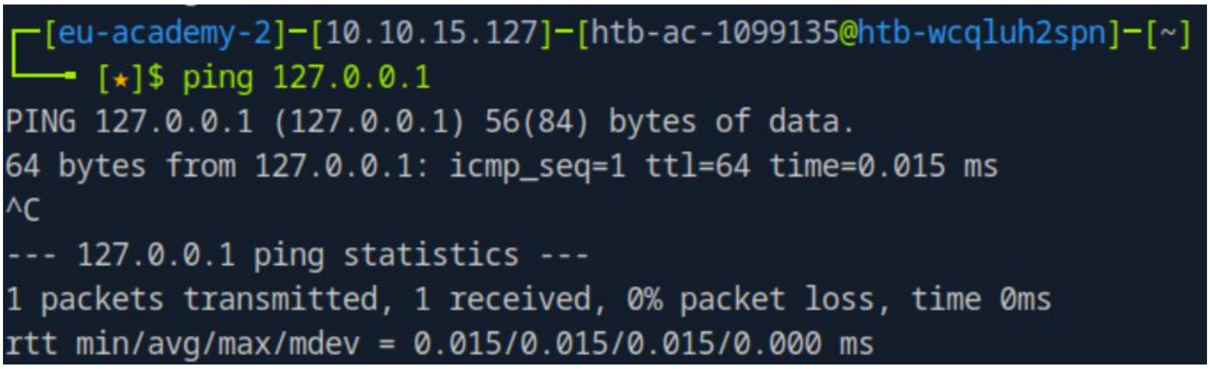
Answer: or

Method:

Question: True or False: TCPDump will resolve IPs to hostnames by default.

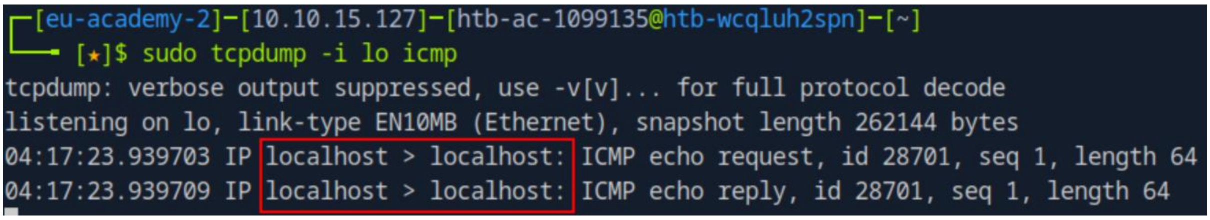
Answer: True

Method: here is a ping sent to localhost (127.0.0.1):



```
┌─[eu-academy-2]─[10.10.15.127]─[htb-ac-1099135@htb-wcqluh2spn]─[~]
└──╼ [★]$ ping 127.0.0.1
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.015 ms
^C
--- 127.0.0.1 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.015/0.015/0.015/0.000 ms
```

And here is the tcpdump capture of which:



```
┌─[eu-academy-2]─[10.10.15.127]─[htb-ac-1099135@htb-wcqluh2spn]─[~]
└──╼ [★]$ sudo tcpdump -i lo icmp
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on lo, link-type EN10MB (Ethernet), snapshot length 262144 bytes
04:17:23.939703 IP localhost > localhost: ICMP echo request, id 28701, seq 1, length 64
04:17:23.939709 IP localhost > localhost: ICMP echo reply, id 28701, seq 1, length 64
```

We can observe that the IP 127.0.0.1 was resolved to the name localhost.

Interrogating Network Traffic With Capture and Display Filters:

Question: What are the client and server port numbers used in first full TCP three-way handshake? (low number first then high number)

Answer: 80 43806

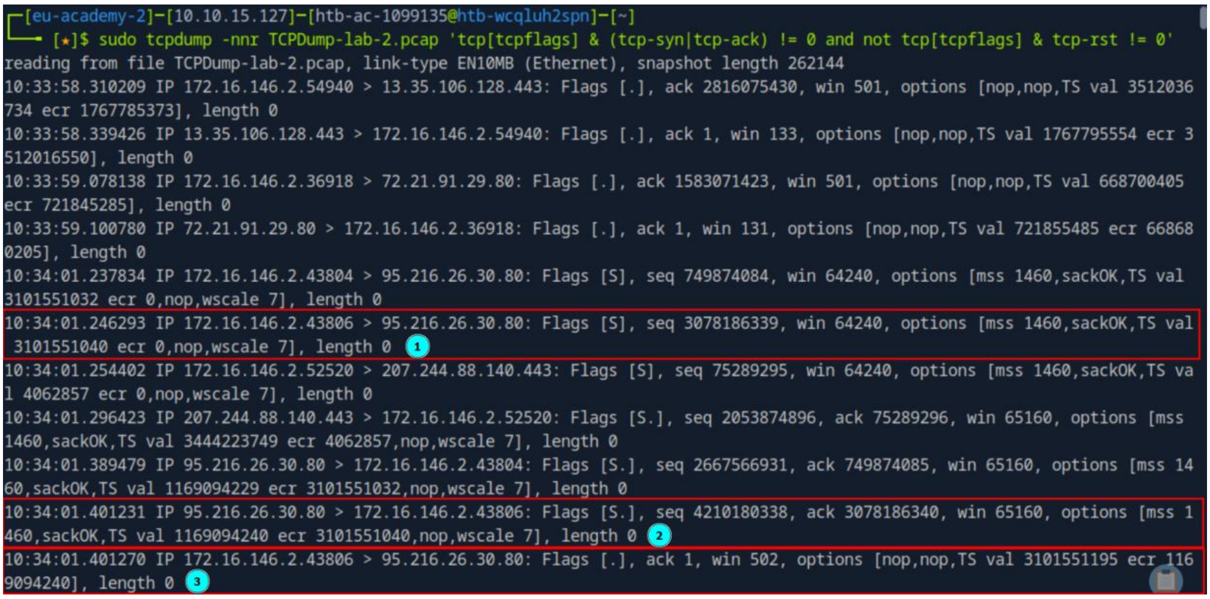
Method: First, we will download to the pwnbox (or any other attacking machine) the ‘TCPDump-Lab-2-Resources’ from the resources bag.

Then – we unzip the file and get a file ‘TCPDump-lab-2.pcap’.

Now, to determine the first complete TCP 3-way-handshake – we will use the following command:

```
sudo tcpdump -nnr TCPDump-lab-2.pcap 'tcp[tcpflags] & (tcp-syn|tcp-ack) != 0 and not tcp[tcpflags] & tcp-rst != 0'
```

which will filter only TCP packets with SYN flag or ACK flag, greatly reduces noise:



```
┌─[eu-academy-2]─[10.10.15.127]─[htb-ac-1099135@htb-wcqluh2spn]─[~]
└──╼ [★]$ sudo tcpdump -nnr TCPDump-lab-2.pcap 'tcp[tcpflags] & (tcp-syn|tcp-ack) != 0 and not tcp[tcpflags] & tcp-rst != 0'
reading from file TCPDump-lab-2.pcap, link-type EN10MB (Ethernet), snapshot length 262144
10:33:58.310209 IP 172.16.146.2.54940 > 13.35.106.128.443: Flags [.], ack 2816075430, win 501, options [nop,nop,TS val 3512036
734 ecr 1767785373], length 0
10:33:58.339426 IP 13.35.106.128.443 > 172.16.146.2.54940: Flags [.], ack 1, win 133, options [nop,nop,TS val 1767795554 ecr 3
512016550], length 0
10:33:59.078138 IP 172.16.146.2.36918 > 72.21.91.29.80: Flags [.], ack 1583071423, win 501, options [nop,nop,TS val 668700405
ecr 721845285], length 0
10:33:59.100780 IP 72.21.91.29.80 > 172.16.146.2.36918: Flags [.], ack 1, win 131, options [nop,nop,TS val 721855485 ecr 66868
0205], length 0
10:34:01.237834 IP 172.16.146.2.43804 > 95.216.26.30.80: Flags [S], seq 749874084, win 64240, options [mss 1460,sackOK,TS val
3101551032 ecr 0,nop,wscale 7], length 0
10:34:01.246293 IP 172.16.146.2.43806 > 95.216.26.30.80: Flags [S], seq 3078186339, win 64240, options [mss 1460,sackOK,TS val
 3101551040 ecr 0,nop,wscale 7], length 0
10:34:01.254402 IP 172.16.146.2.52520 > 207.244.88.140.443: Flags [S], seq 75289295, win 64240, options [mss 1460,sackOK,TS va
l 4062857 ecr 0,nop,wscale 7], length 0
10:34:01.296423 IP 207.244.88.140.443 > 172.16.146.2.52520: Flags [S.], seq 2053874896, ack 75289296, win 65160, options [mss
1460,sackOK,TS val 3444223749 ecr 4062857,nop,wscale 7], length 0
10:34:01.389479 IP 95.216.26.30.80 > 172.16.146.2.43804: Flags [S.], seq 2667566931, ack 749874085, win 65160, options [mss 14
60,sackOK,TS val 1169094229 ecr 3101551032,nop,wscale 7], length 0
10:34:01.401231 IP 95.216.26.30.80 > 172.16.146.2.43806: Flags [S.], seq 4210180338, ack 3078186340, win 65160, options [mss 1
460,sackOK,TS val 1169094240 ecr 3101551040,nop,wscale 7], length 0
10:34:01.401270 IP 172.16.146.2.43806 > 95.216.26.30.80: Flags [.], ack 1, win 502, options [nop,nop,TS val 3101551195 ecr 116
9094240], length 0
```

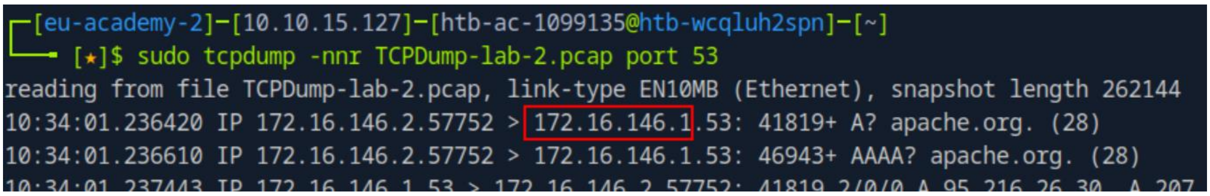
The marked packets constitute the TCP 3-way handshake – where the first marked packet is the SYN, the 2nd packet is the SYN-ACK, and the 3rd packet is the ACK. We can also see that is that first complete 3-way handshake.

Question: Based on the traffic seen in the pcap file, who is the DNS server in this network segment? (ip address)

Answer: 172.16.146.1

Method: we read the pcap traffic, filtering for port 53 (DNS):

```
sudo tcpdump -nnr TCPDump-lab-2.pcap port 53
```



```
┌─[eu-academy-2]─[10.10.15.127]─[htb-ac-1099135@htb-wcqluh2spn]─[~]
└──╼ [★]$ sudo tcpdump -nnr TCPDump-lab-2.pcap port 53
reading from file TCPDump-lab-2.pcap, link-type EN10MB (Ethernet), snapshot length 262144
10:34:01.236420 IP 172.16.146.2.57752 > 172.16.146.1.53: 41819+ A? apache.org. (28)
10:34:01.236610 IP 172.16.146.2.57752 > 172.16.146.1.53: 46943+ AAAA? apache.org. (28)
10:34:01.237443 IP 172.16.146.1.53 > 172.16.146.2.57752: 41819 2/0/0 A 95.216.26.30, A 207
```

And we can see the DNS server in the first packet.

Wireshark

Analysis with Wireshark:

Question: True or False: Wireshark can run on both Windows and Linux.

Answer: True

Method:

Question: Which Pane allows a user to see a summary of each packet grabbed during the capture?

Answer: Packet List

Method:

Question: Which pane provides you insight into the traffic you captured and displays it in both ASCII and Hex?

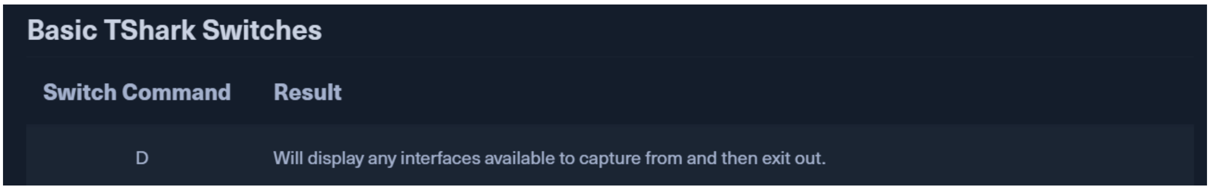
Answer: Packet Bytes

Method:

Question: What switch is used with TShark to list possible interfaces to capture on?

Answer: -D

Method:



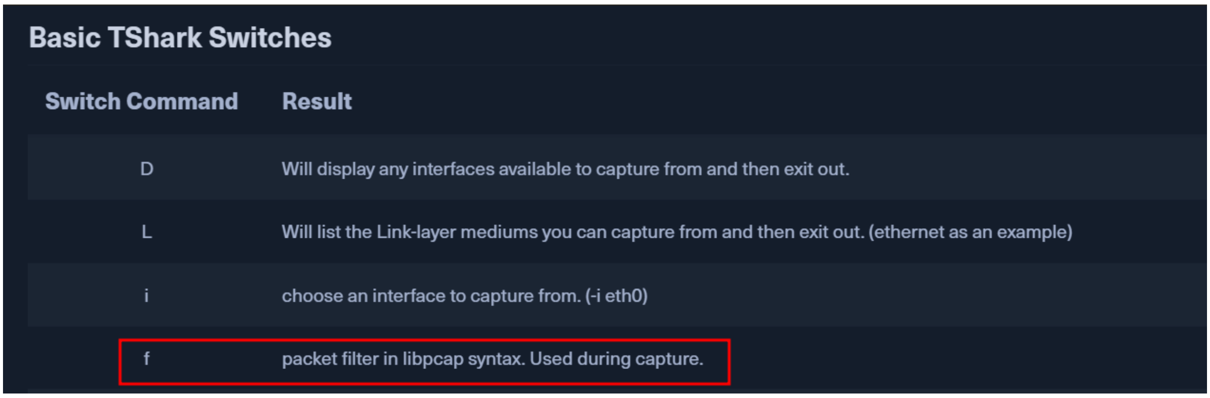
Basic TShark Switches

Switch Command	Result
D	Will display any interfaces available to capture from and then exit out.

Question: What switch allows us to apply filters in TShark?

Answer: -f

Method:



Basic TShark Switches

Switch Command	Result
D	Will display any interfaces available to capture from and then exit out.
L	Will list the Link-layer mediums you can capture from and then exit out. (ethernet as an example)
i	choose an interface to capture from. (-i eth0)
f	packet filter in libpcap syntax. Used during capture.

Question: Is a capture filter applied before the capture starts or after? (answer before or after)

Answer: before

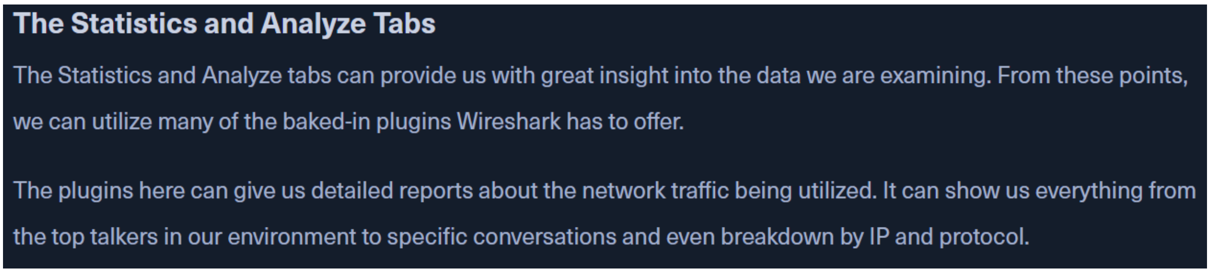
Method: 'While capturing traffic with Wireshark, we have several options regarding how and when we filter out traffic. This is accomplished utilizing Capture and Display filters. The Former initiated before the capture starts and the latter during or after capture is complete.'

Wireshark Advanced Usage:

Question: Which plugin tab can provide us with a way to view conversation metadata and even protocol breakdowns for the entire PCAP file?

Answer: Statistics

Method:



The Statistics and Analyze Tabs

The Statistics and Analyze tabs can provide us with great insight into the data we are examining. From these points, we can utilize many of the baked-in plugins Wireshark has to offer.

The plugins here can give us detailed reports about the network traffic being utilized. It can show us everything from the top talkers in our environment to specific conversations and even breakdown by IP and protocol.

Question: What plugin tab will allow me to accomplish tasks such as applying filters, following streams, and viewing expert info?

Answer: Analyze

Method: ‘From the Analyze tab, we can utilize plugins that allow us to do things such as following TCP streams, filter on conversation types, prepare new packet filters and examine the expert info Wireshark generates about the traffic.’

Question: What stream oriented Transport protocol enables us to follow and rebuild conversations and the included data?

Answer: TCP

Method: ‘Wireshark can stitch TCP packets back together to recreate the entire stream in a readable format. This ability also allows us to pull data (images, files, etc.) out of the capture. This works for almost any protocol that utilizes TCP as a transport mechanism.’

Question: True or False: Wireshark can extract files from HTTP traffic.

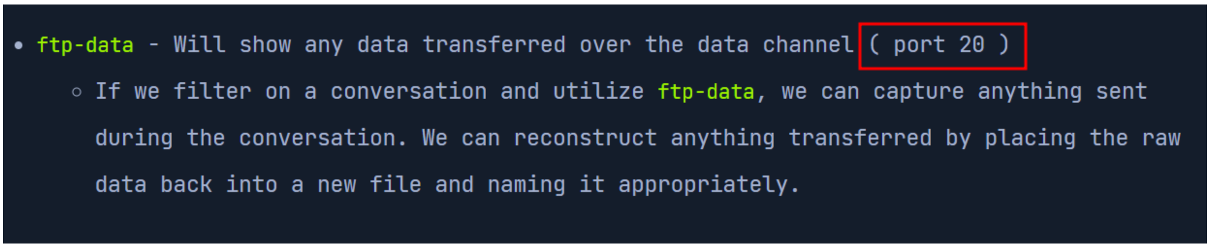
Answer: True

Method:

Question: True or False: The ftp-data filter will show us any data sent over TCP port 21.

Answer: False

Method: the mentioned port is port 20:



- ftp-data - Will show any data transferred over the data channel (port 20)
  - If we filter on a conversation and utilize ftp-data, we can capture anything sent during the conversation. We can reconstruct anything transferred by placing the raw data back into a new file and naming it appropriately.

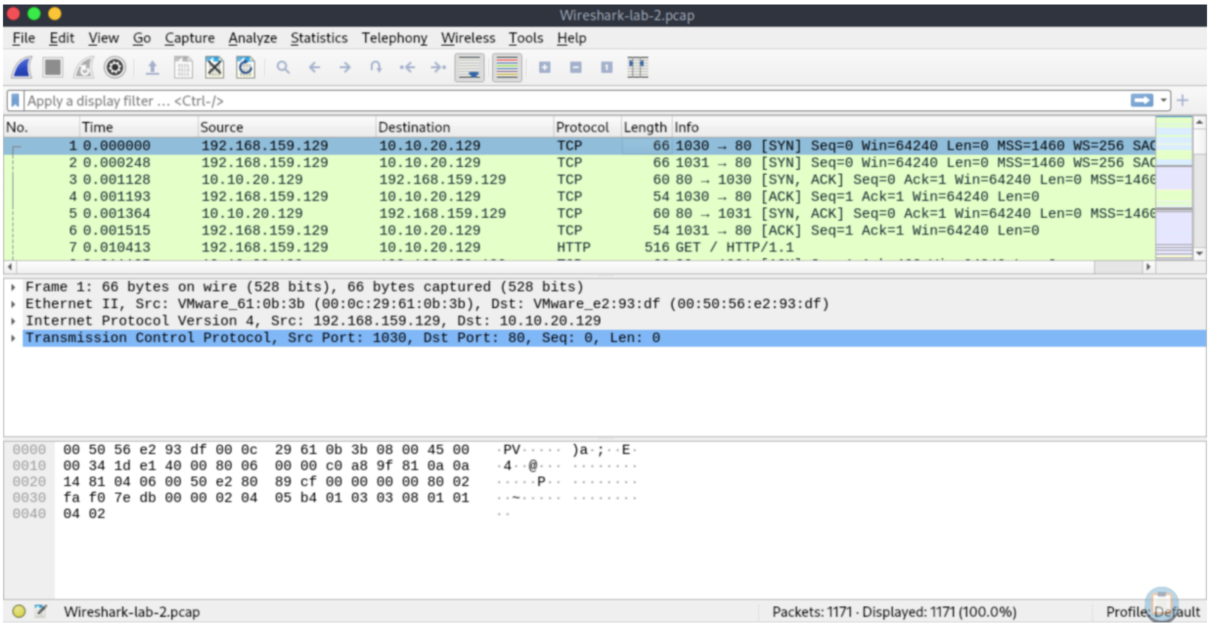
Packet Inception, Dissecting Network Traffic With Wireshark:

Question: What was the filename of the image that contained a certain Transformer Leader? (name.filetype)

Answer: Rise-Up.jpg

Method: First, lets download the 'Wireshark-Lab-2-Resources' from the resources bag to the pwnbox.

Then - we unzip it, and open the extracted file 'Wireshark-lab-2.pcap' with Wireshark.



Wireshark-lab-2.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>

```
No.   Time        Source            Destination       Protocol Length Info
    1 0.000000    192.168.159.129   10.10.20.129      TCP         66 1030 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SAC
    2 0.000248    192.168.159.129   10.10.20.129      TCP         66 1031 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SAC
    3 0.001128    10.10.20.129      192.168.159.129   TCP         60 80 → 1030 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=146
    4 0.001193    192.168.159.129   10.10.20.129      TCP         54 1030 → 80 [ACK] Seq=1 Ack=1 Win=64240 Len=0
    5 0.001364    10.10.20.129      192.168.159.129   TCP         60 80 → 1031 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=146
    6 0.001515    192.168.159.129   10.10.20.129      TCP         54 1031 → 80 [ACK] Seq=1 Ack=1 Win=64240 Len=0
    7 0.010413    192.168.159.129   10.10.20.129      HTTP       516 GET / HTTP/1.1
```

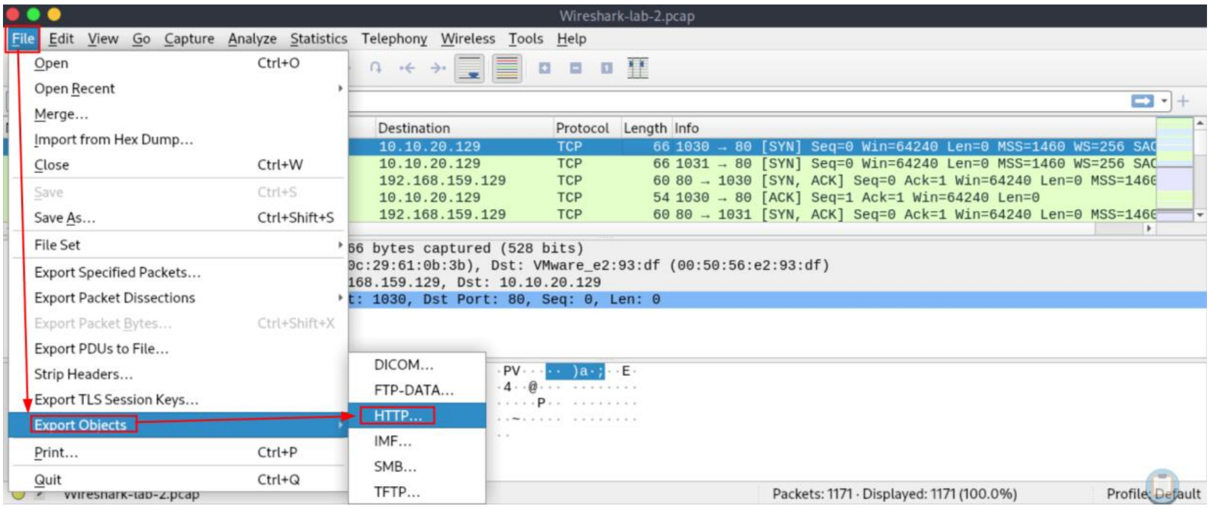
```
▸ Frame 1: 66 bytes on wire (528 bits), 66 bytes captured (528 bits)
▸ Ethernet II, Src: VMware_61:0b:3b (00:0c:29:61:0b:3b), Dst: VMware_e2:93:df (00:50:56:e2:93:df)
▸ Internet Protocol Version 4, Src: 192.168.159.129, Dst: 10.10.20.129
▸ Transmission Control Protocol, Src Port: 1030, Dst Port: 80, Seq: 0, Len: 0
```

```
0000  00 50 56 e2 93 df 00 0c  29 61 0b 3b 08 00 45 00   ·PV·····)a·;··E·
0010  00 34 1d e1 40 00 80 06  00 00 c0 a8 9f 81 0a 0a   ·4··@···········
0020  14 81 04 06 00 50 e2 80  89 cf 00 00 00 00 80 02   ·····P··········
0030  fa f0 7e db 00 00 02 04  05 b4 01 03 03 08 01 01   ··~·············
0040  04 02                                              ··
```

Wireshark-lab-2.pcap Packets: 1171 · Displayed: 1171 (100.0%) Profile: Default

We can see there is a plenty HTTP traffic.

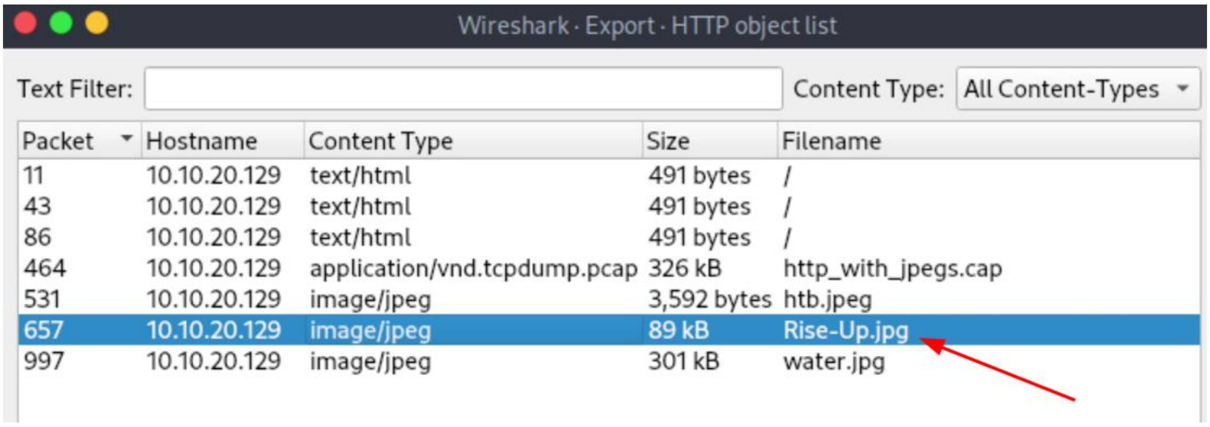
Now where the wireshark pcap recording is open – we obtain the file name in 'File' → 'Export Objects' → 'HTTP':



```
Destination      Protocol Length Info
10.10.20.129     TCP        66 1030 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SAC
10.10.20.129     TCP        66 1031 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SAC
192.168.159.129  TCP        60 80 → 1030 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=146
10.10.20.129     TCP        54 1030 → 80 [ACK] Seq=1 Ack=1 Win=64240 Len=0
192.168.159.129  TCP        60 80 → 1031 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=146

66 bytes captured (528 bits)
0c:29:61:0b:3b), Dst: VMware_e2:93:df (00:50:56:e2:93:df)
168.159.129, Dst: 10.10.20.129
t: 1030, Dst Port: 80, Seq: 0, Len: 0

·PV· · ·· · )a·;· ·E·
·4· ·@· · · · · · · · ·
· · · · ·P· · · · · · · · ·
· ·~· · · · · · · · · · ·
```

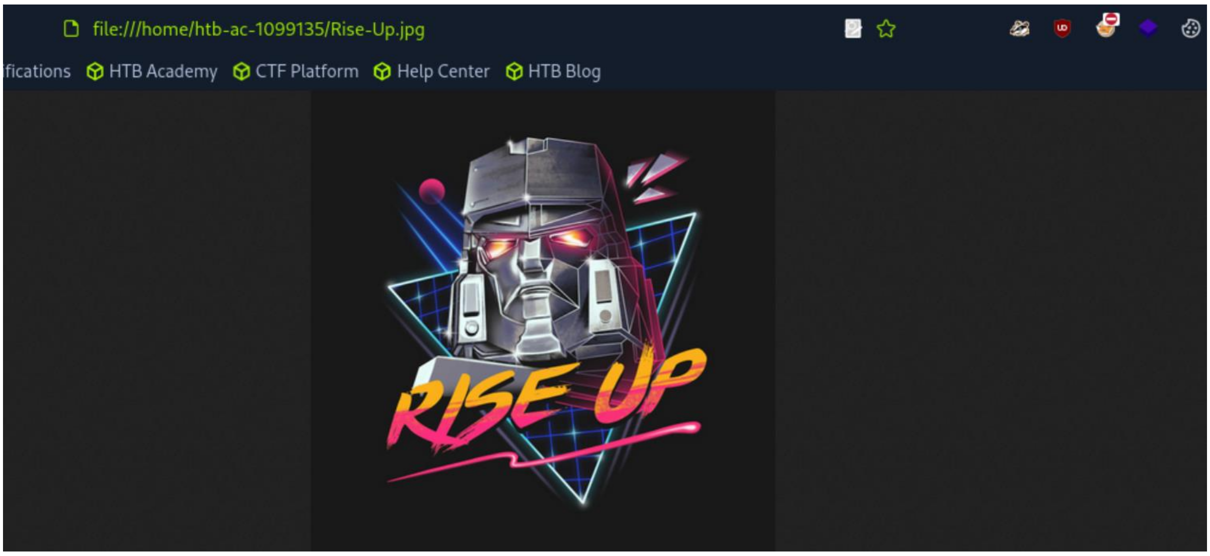
A widow with the HTTP object, including the images is opened.

Heeding to the hint – we use the process of elimination to determine the image of the transformers leader:


Wireshark · Export · HTTP object list
Text Filter:
Content Type: All Content-Types

Packet	Hostname	Content Type	Size	Filename
11	10.10.20.129	text/html	491 bytes	/
43	10.10.20.129	text/html	491 bytes	/
86	10.10.20.129	text/html	491 bytes	/
464	10.10.20.129	application/vnd.tcpdump.pcap	326 kB	http_with_jpegs.cap
531	10.10.20.129	image/jpeg	3,592 bytes	htb.jpeg
657	10.10.20.129	image/jpeg	89 kB	Rise-Up.jpg
997	10.10.20.129	image/jpeg	301 kB	water.jpg

We can continue to save it, and open the file with the browser:


file:///home/htb-ac-1099135/Rise-Up.jpg

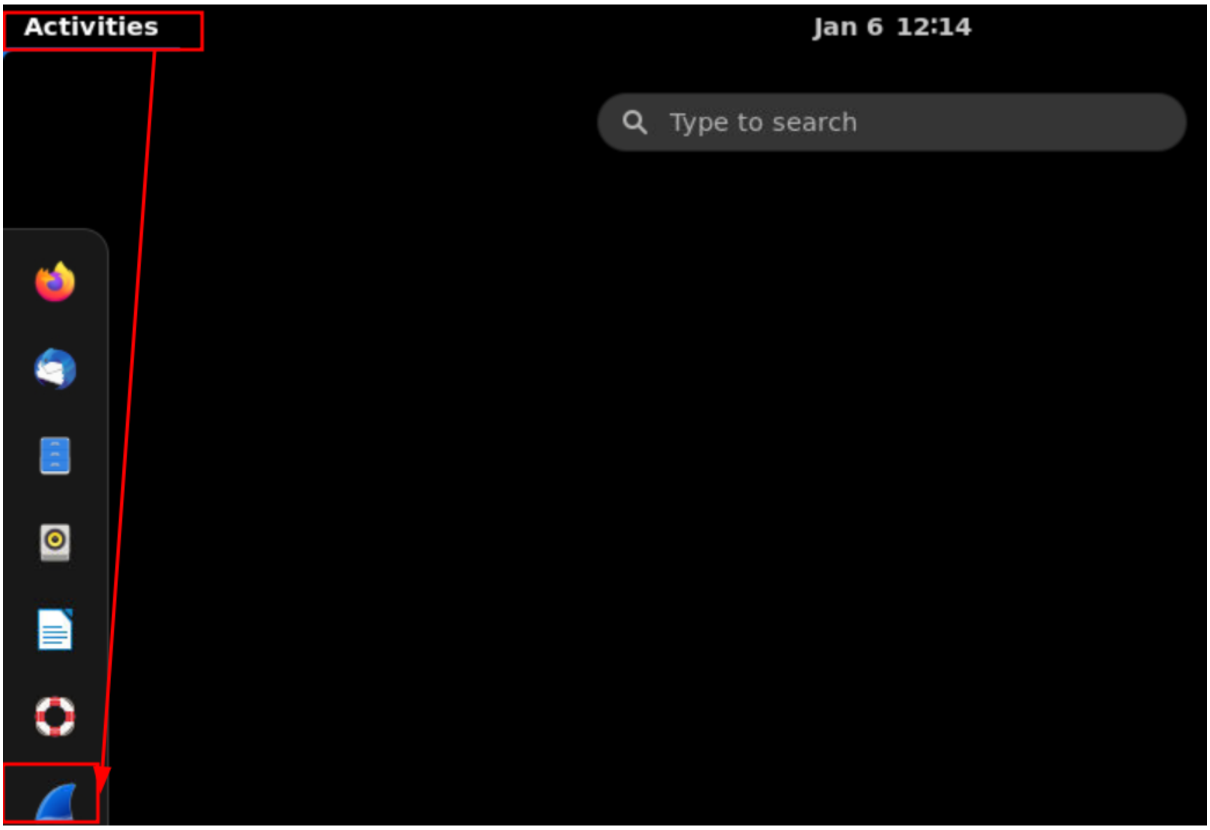
Question: Which employee is suspected of performing potentially malicious actions in the live environment?

Answer: bob

Method: First, lets RDP to the target machine with the provided credentials: 'htb-student:HTB_@cademy_stdnt!'

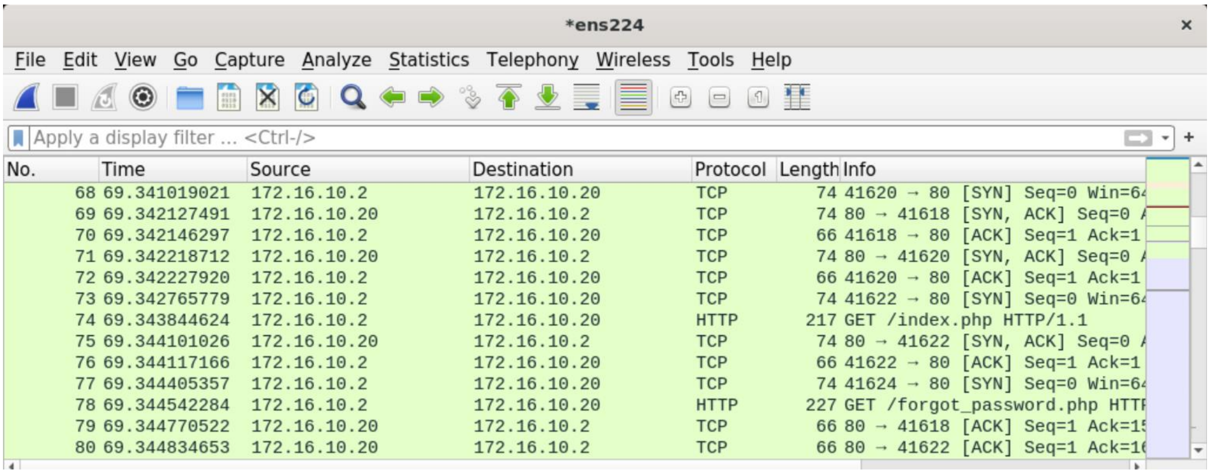
```
xfreerdp /v:<Target IP> /u:htb-student /p:HTB_@cademy_stdnt!
/dynamic-resolution
```

and in the RDP session – we open wireshark:



And start capture on interface 'ens224'.

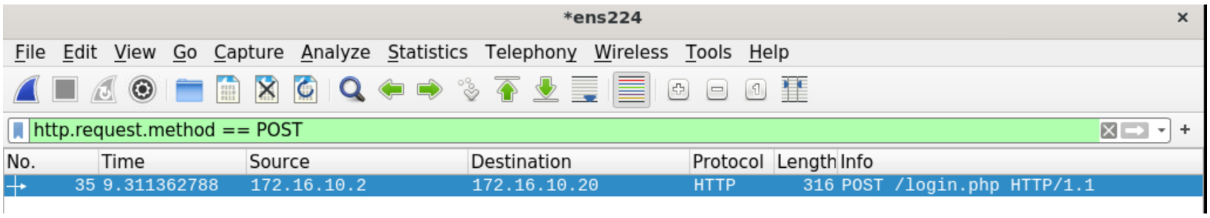
After some time – we will get a lot of traffic, including HTTP traffic:


*ens224
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>

```
No.   Time          Source        Destination   Protocol Length Info
   68 69.341019021  172.16.10.2   172.16.10.20  TCP        74 41620 → 80 [SYN] Seq=0 Win=64
   69 69.342127491  172.16.10.20  172.16.10.2   TCP        74 80 → 41618 [SYN, ACK] Seq=0 A
   70 69.342146297  172.16.10.2   172.16.10.20  TCP        66 41618 → 80 [ACK] Seq=1 Ack=1
   71 69.342218712  172.16.10.20  172.16.10.2   TCP        74 80 → 41620 [SYN, ACK] Seq=0 A
   72 69.342227920  172.16.10.2   172.16.10.20  TCP        66 41620 → 80 [ACK] Seq=1 Ack=1
   73 69.342765779  172.16.10.2   172.16.10.20  TCP        74 41622 → 80 [SYN] Seq=0 Win=64
   74 69.343844624  172.16.10.2   172.16.10.20  HTTP      217 GET /index.php HTTP/1.1
   75 69.344101026  172.16.10.20  172.16.10.2   TCP        74 80 → 41622 [SYN, ACK] Seq=0 A
   76 69.344117166  172.16.10.2   172.16.10.20  TCP        66 41622 → 80 [ACK] Seq=1 Ack=1
   77 69.344405357  172.16.10.2   172.16.10.20  TCP        74 41624 → 80 [SYN] Seq=0 Win=64
   78 69.344542284  172.16.10.2   172.16.10.20  HTTP      227 GET /forgot_password.php HTTP
   79 69.344770522  172.16.10.20  172.16.10.2   TCP        66 80 → 41618 [ACK] Seq=1 Ack=15
   80 69.344834653  172.16.10.20  172.16.10.2   TCP        66 80 → 41622 [ACK] Seq=1 Ack=16
```

We will have to reduce the noise – so lets filter for POST requests:

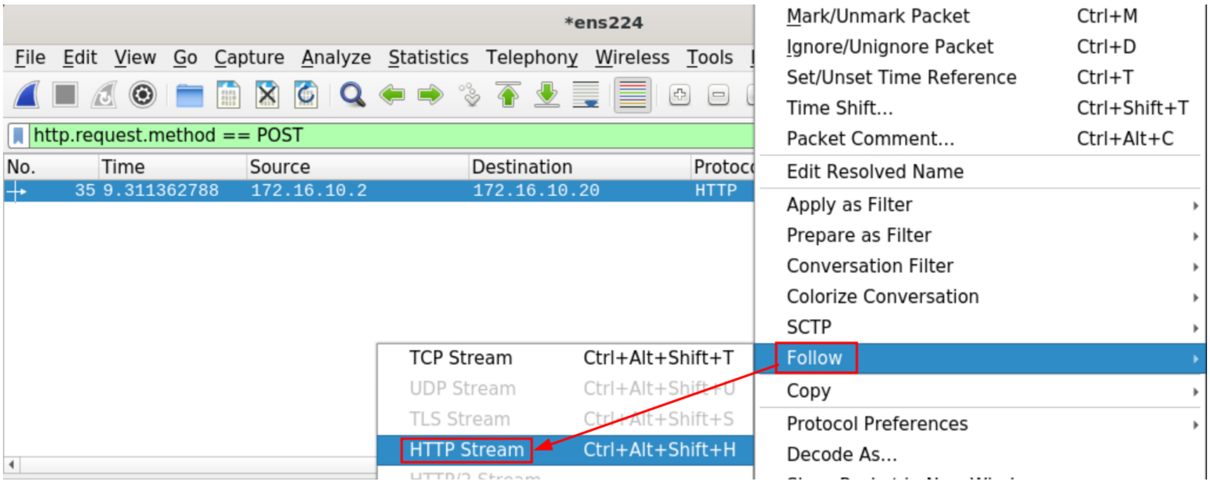
```
http.request.method == POST
```


*ens224
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

```
http.request.method == POST
No.   Time          Source        Destination   Protocol Length Info
   35 9.311362788   172.16.10.2   172.16.10.20  HTTP      316 POST /login.php HTTP/1.1
```

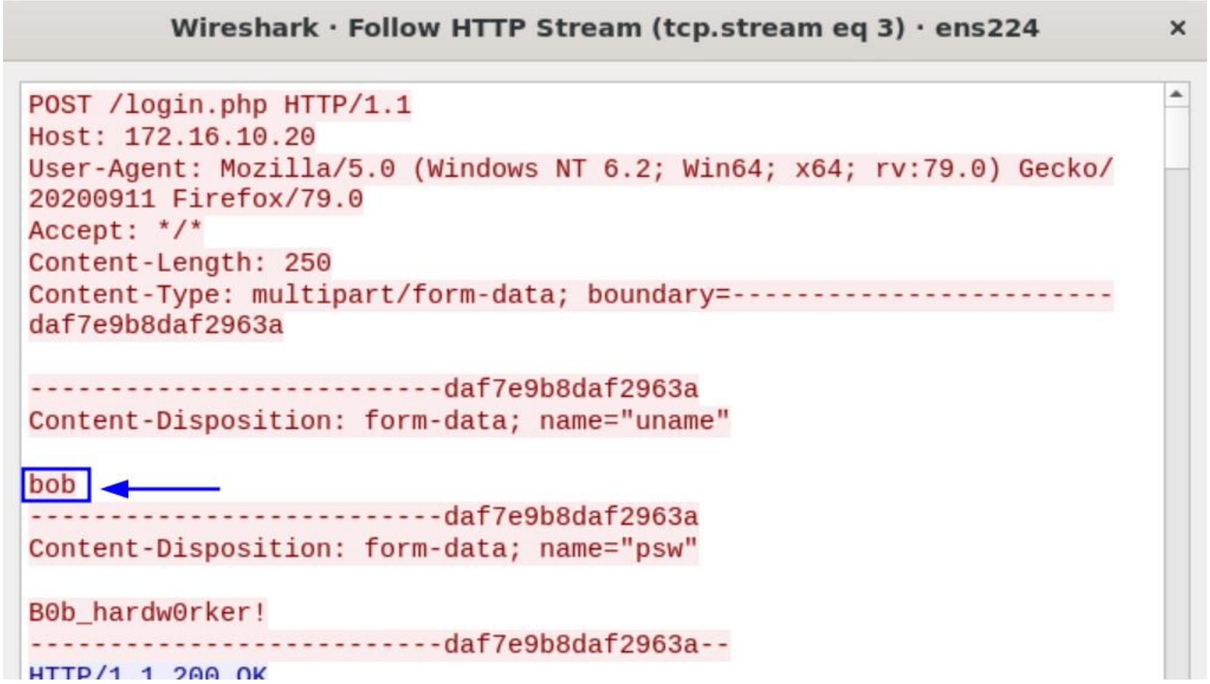
There is this one login POST request, that may include credentials.

Lets inspect it further using HTTP stream, right clicking the packet:


*ens224
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools
Mark/Unmark Packet Ctrl+M
Ignore/Unignore Packet Ctrl+D
Set/Unset Time Reference Ctrl+T
Time Shift… Ctrl+Shift+T
Packet Comment… Ctrl+Alt+C
Edit Resolved Name
Apply as Filter
Prepare as Filter
Conversation Filter
Colorize Conversation
SCTP
Follow
Copy
Protocol Preferences
Decode As…
TCP Stream Ctrl+Alt+Shift+T
UDP Stream Ctrl+Alt+Shift+U
TLS Stream Ctrl+Alt+Shift+S
HTTP Stream Ctrl+Alt+Shift+H
HTTP/2 Stream

```
http.request.method == POST
No.   Time          Source        Destination   Protocol
   35 9.311362788   172.16.10.2   172.16.10.20  HTTP
```

And in it we can see the username entered, and his password:



Wireshark · Follow HTTP Stream (tcp.stream eq 3) · ens224

```
POST /login.php HTTP/1.1
Host: 172.16.10.20
User-Agent: Mozilla/5.0 (Windows NT 6.2; Win64; x64; rv:79.0) Gecko/
20200911 Firefox/79.0
Accept: */*
Content-Length: 250
Content-Type: multipart/form-data; boundary=------------------------
daf7e9b8daf2963a

--------------------------daf7e9b8daf2963a
Content-Disposition: form-data; name="uname"

bob
--------------------------daf7e9b8daf2963a
Content-Disposition: form-data; name="psw"

B0b_hardw0rker!
--------------------------daf7e9b8daf2963a--
HTTP/1.1 200 OK
```

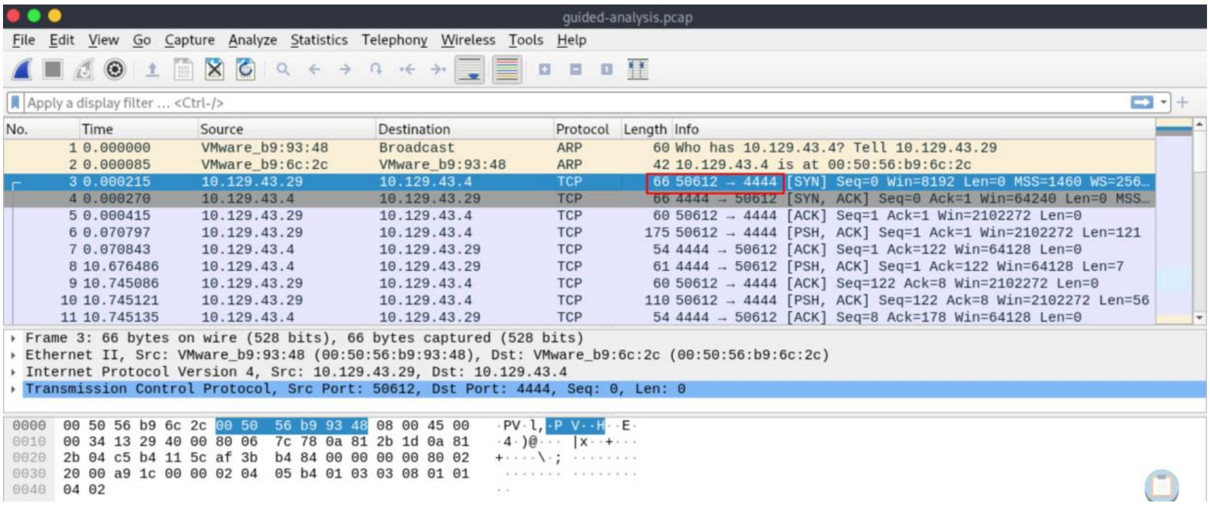
Guided Lab: Traffic Analysis Workflow:

Question: What was the name of the new user created on mrb3n's host?

Answer: hacker

Method: First, lets download the 'Guided-Analysis-Lab-Resources' from the resources bag to the pwnbox.

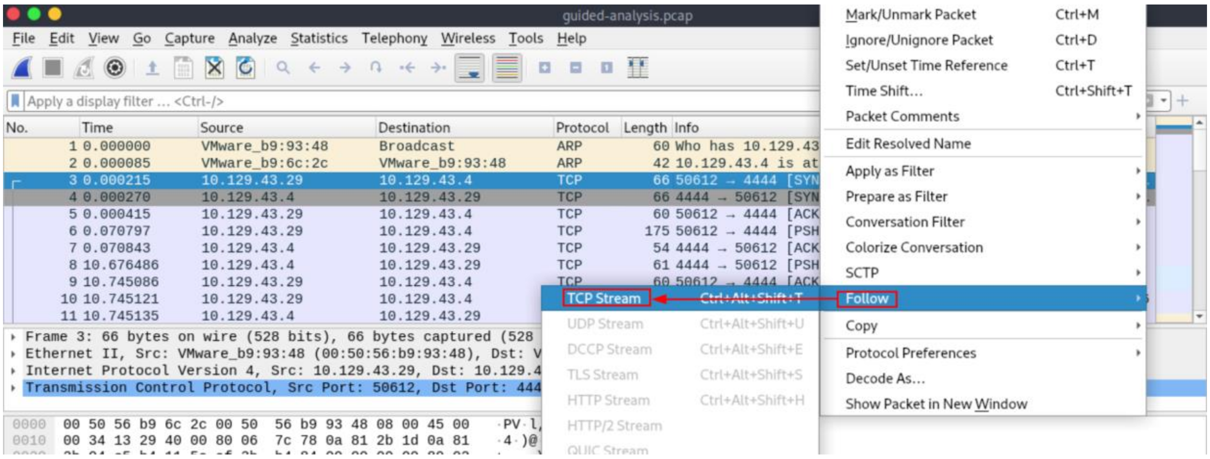
Then - we unzip it, and open the extracted file 'guided-analysis.pcap' with Wireshark.


guided-analysis.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>

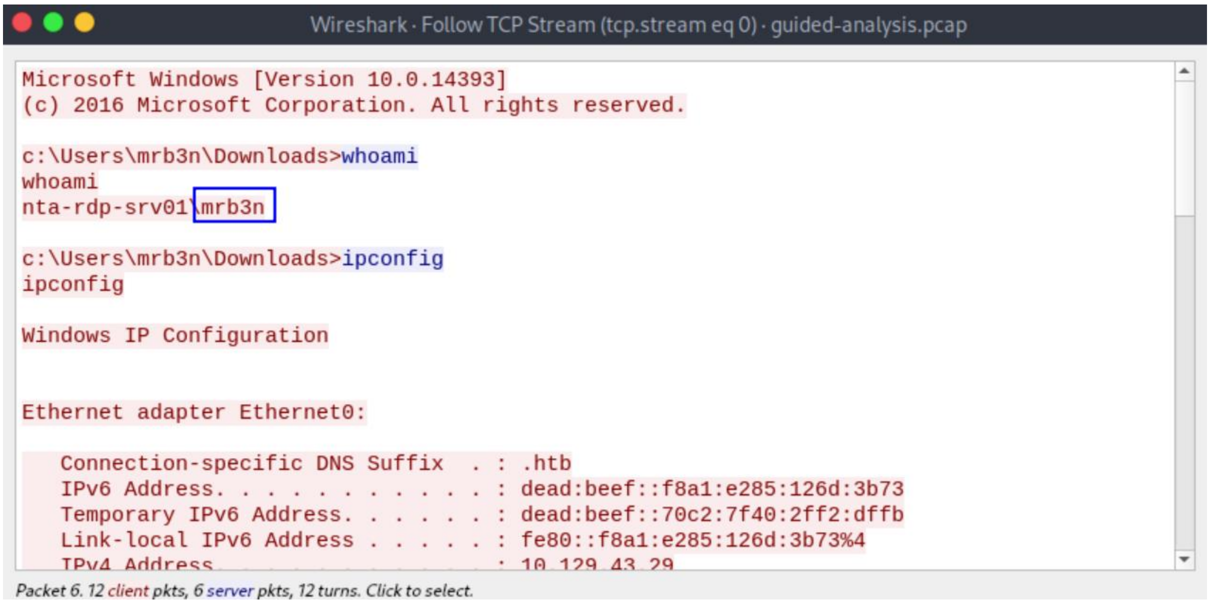
```
No.  Time        Source            Destination       Protocol Length Info
   1 0.000000    VMware_b9:93:48   Broadcast         ARP        60 Who has 10.129.43.4? Tell 10.129.43.29
   2 0.000085    VMware_b9:6c:2c   VMware_b9:93:48   ARP        42 10.129.43.4 is at 00:50:56:b9:6c:2c
   3 0.000215    10.129.43.29      10.129.43.4       TCP        66 50612 → 4444 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256…
   4 0.000270    10.129.43.4       10.129.43.29      TCP        66 4444 → 50612 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS…
   5 0.000415    10.129.43.29      10.129.43.4       TCP        60 50612 → 4444 [ACK] Seq=1 Ack=1 Win=2102272 Len=0
   6 0.070797    10.129.43.29      10.129.43.4       TCP       175 50612 → 4444 [PSH, ACK] Seq=1 Ack=1 Win=2102272 Len=121
   7 0.070843    10.129.43.4       10.129.43.29      TCP        54 4444 → 50612 [ACK] Seq=1 Ack=122 Win=64128 Len=0
   8 10.676486   10.129.43.4       10.129.43.29      TCP        61 4444 → 50612 [PSH, ACK] Seq=1 Ack=122 Win=64128 Len=7
   9 10.745086   10.129.43.29      10.129.43.4       TCP        60 50612 → 4444 [ACK] Seq=122 Ack=8 Win=2102272 Len=0
  10 10.745121   10.129.43.29      10.129.43.4       TCP       110 50612 → 4444 [PSH, ACK] Seq=122 Ack=8 Win=2102272 Len=56
  11 10.745135   10.129.43.4       10.129.43.29      TCP        54 4444 → 50612 [ACK] Seq=8 Ack=178 Win=64128 Len=0

▸ Frame 3: 66 bytes on wire (528 bits), 66 bytes captured (528 bits)
▸ Ethernet II, Src: VMware_b9:93:48 (00:50:56:b9:93:48), Dst: VMware_b9:6c:2c (00:50:56:b9:6c:2c)
▸ Internet Protocol Version 4, Src: 10.129.43.29, Dst: 10.129.43.4
▸ Transmission Control Protocol, Src Port: 50612, Dst Port: 4444, Seq: 0, Len: 0

0000  00 50 56 b9 6c 2c 00 50  56 b9 93 48 08 00 45 00   ·PV·l,·P V··H··E·
0010  00 34 13 29 40 00 80 06  7c 78 0a 81 2b 1d 0a 81   ·4·)@··· |x··+···
0020  2b 04 c5 b4 11 5c af 3b  b4 84 00 00 00 00 80 02   +····\·; ········
0030  20 00 a9 1c 00 00 02 04  05 b4 01 03 03 08 01 01    ······· ········
0040  04 02                                              ··
```

We can see TCP traffic to port 4444.

We will follow the packet's TCP stream:


guided-analysis.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>
Mark/Unmark Packet Ctrl+M
Ignore/Unignore Packet 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
TCP Stream Ctrl+Alt+Shift+T
UDP Stream Ctrl+Alt+Shift+U
DCCP Stream Ctrl+Alt+Shift+E
TLS Stream Ctrl+Alt+Shift+S
HTTP Stream Ctrl+Alt+Shift+H
HTTP/2 Stream
QUIC Stream

```
No.  Time        Source            Destination       Protocol Length Info
   1 0.000000    VMware_b9:93:48   Broadcast         ARP        60 Who has 10.129.43
   2 0.000085    VMware_b9:6c:2c   VMware_b9:93:48   ARP        42 10.129.43.4 is at
   3 0.000215    10.129.43.29      10.129.43.4       TCP        66 50612 → 4444 [SYN
   4 0.000270    10.129.43.4       10.129.43.29      TCP        66 4444 → 50612 [SYN
   5 0.000415    10.129.43.29      10.129.43.4       TCP        60 50612 → 4444 [ACK
   6 0.070797    10.129.43.29      10.129.43.4       TCP       175 50612 → 4444 [PSH
   7 0.070843    10.129.43.4       10.129.43.29      TCP        54 4444 → 50612 [ACK
   8 10.676486   10.129.43.4       10.129.43.29      TCP        61 4444 → 50612 [PSH
   9 10.745086   10.129.43.29      10.129.43.4       TCP        60 50612 → 4444 [ACK
  10 10.745121   10.129.43.29      10.129.43.4
  11 10.745135   10.129.43.4       10.129.43.29

▸ Frame 3: 66 bytes on wire (528 bits), 66 bytes captured (528
▸ Ethernet II, Src: VMware_b9:93:48 (00:50:56:b9:93:48), Dst: V
▸ Internet Protocol Version 4, Src: 10.129.43.29, Dst: 10.129.4
▸ Transmission Control Protocol, Src Port: 50612, Dst Port: 444

0000  00 50 56 b9 6c 2c 00 50  56 b9 93 48 08 00 45 00   ·PV·l,
0010  00 34 13 29 40 00 80 06  7c 78 0a 81 2b 1d 0a 81   ·4·)@
```



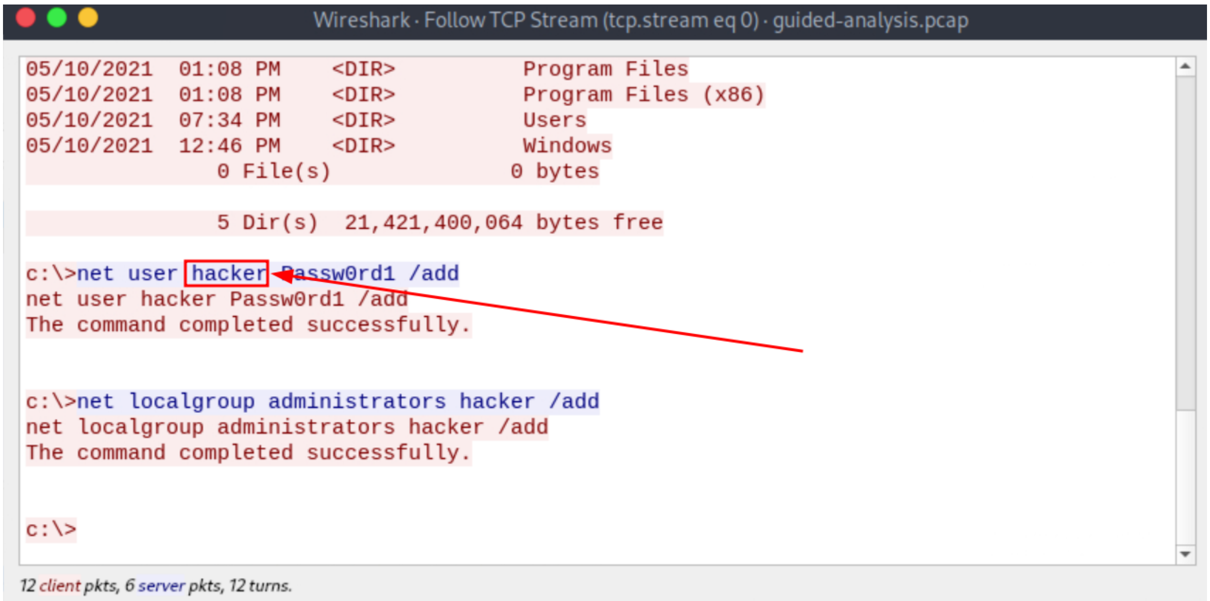
Wireshark · Follow TCP Stream (tcp.stream eq 0) · guided-analysis.pcap

```
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

c:\Users\mrb3n\Downloads>whoami
whoami
nta-rdp-srv01\mrb3n

c:\Users\mrb3n\Downloads>ipconfig
ipconfig

Windows IP Configuration


Ethernet adapter Ethernet0:

   Connection-specific DNS Suffix  . : .htb
   IPv6 Address. . . . . . . . . . . : dead:beef::f8a1:e285:126d:3b73
   Temporary IPv6 Address. . . . . . : dead:beef::70c2:7f40:2ff2:dffb
   Link-local IPv6 Address . . . . . : fe80::f8a1:e285:126d:3b73%4
   IPv4 Address. . . . . . . . . . . : 10.129.43.29
```

Packet 6. 12 client pkts, 6 server pkts, 12 turns. Click to select.

Stram reveals that is indeed ‘mrb3n’ host, with a shell on it.

Scrolling down:



Wireshark · Follow TCP Stream (tcp.stream eq 0) · guided-analysis.pcap

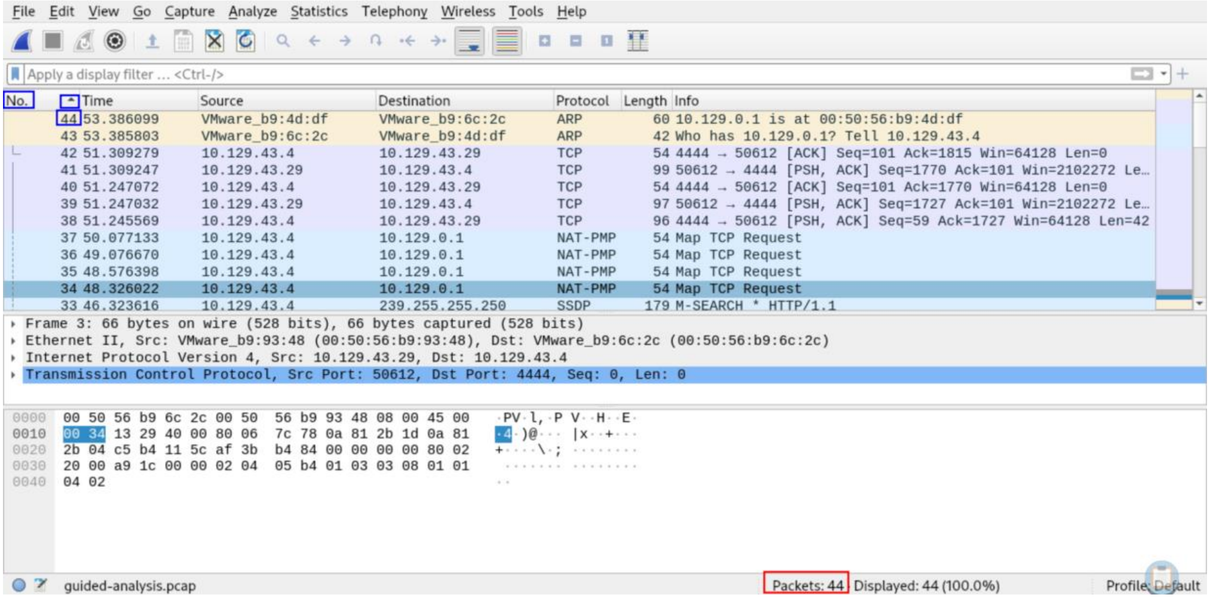
```
05/10/2021  01:08 PM    <DIR>          Program Files
05/10/2021  01:08 PM    <DIR>          Program Files (x86)
05/10/2021  07:34 PM    <DIR>          Users
05/10/2021  12:46 PM    <DIR>          Windows
               0 File(s)              0 bytes

               5 Dir(s)  21,421,400,064 bytes free

c:\>net user hacker Passw0rd1 /add
net user hacker Passw0rd1 /add
The command completed successfully.


c:\>net localgroup administrators hacker /add
net localgroup administrators hacker /add
The command completed successfully.


c:\>
```

The created user on it is ‘hacker’

Question: How many total packets were there in the Guided-analysis PCAP?

Answer: 44

Method:


File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
44	53.386099	VMware_b9:4d:df	VMware_b9:6c:2c	ARP	60	10.129.0.1 is at 00:50:56:b9:4d:df
43	53.385803	VMware_b9:6c:2c	VMware_b9:4d:df	ARP	42	Who has 10.129.0.1? Tell 10.129.43.4
42	51.309279	10.129.43.4	10.129.43.29	TCP	54	4444 → 50612 [ACK] Seq=101 Ack=1815 Win=64128 Len=0
41	51.309247	10.129.43.29	10.129.43.4	TCP	99	50612 → 4444 [PSH, ACK] Seq=1770 Ack=101 Win=2102272 Le…
40	51.247072	10.129.43.4	10.129.43.29	TCP	54	4444 → 50612 [ACK] Seq=101 Ack=1770 Win=64128 Len=0
39	51.247032	10.129.43.29	10.129.43.4	TCP	97	50612 → 4444 [PSH, ACK] Seq=1727 Ack=101 Win=2102272 Le…
38	51.245569	10.129.43.4	10.129.43.29	TCP	96	4444 → 50612 [PSH, ACK] Seq=59 Ack=1727 Win=64128 Len=42
37	50.077133	10.129.43.4	10.129.0.1	NAT-PMP	54	Map TCP Request
36	49.076670	10.129.43.4	10.129.0.1	NAT-PMP	54	Map TCP Request
35	48.576398	10.129.43.4	10.129.0.1	NAT-PMP	54	Map TCP Request
34	48.326022	10.129.43.4	10.129.0.1	NAT-PMP	54	Map TCP Request
33	46.323616	10.129.43.4	239.255.255.250	SSDP	179	M-SEARCH * HTTP/1.1

```
▸ Frame 3: 66 bytes on wire (528 bits), 66 bytes captured (528 bits)
▸ Ethernet II, Src: VMware_b9:93:48 (00:50:56:b9:93:48), Dst: VMware_b9:6c:2c (00:50:56:b9:6c:2c)
▸ Internet Protocol Version 4, Src: 10.129.43.29, Dst: 10.129.43.4
▸ Transmission Control Protocol, Src Port: 50612, Dst Port: 4444, Seq: 0, Len: 0
```

```
0000  00 50 56 b9 6c 2c 00 50  56 b9 93 48 08 00 45 00   ·PV·l,·P V··H··E·
0010  00 34 13 29 40 00 80 06  7c 78 0a 81 2b 1d 0a 81   ·4·)@··· |x··+···
0020  2b 04 c5 b4 11 5c af 3b  b4 84 00 00 00 00 80 02   +····\·; ········
0030  20 00 a9 1c 00 00 02 04  05 b4 01 03 03 08 01 01    ······· ········
0040  04 02                                              ··
```

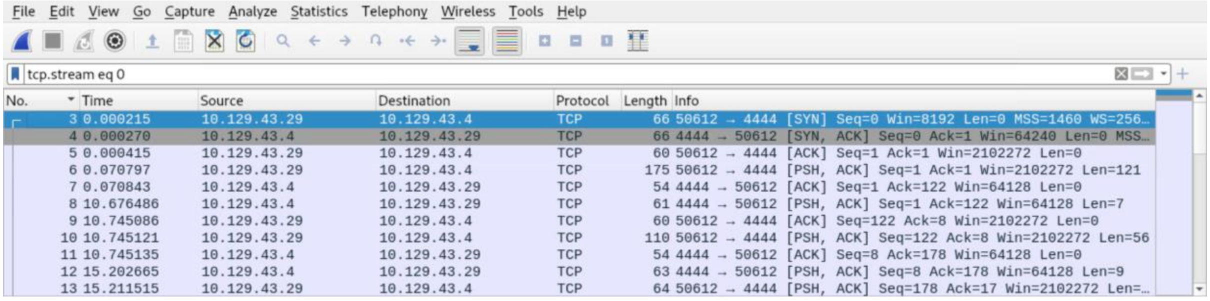
Method 1 (marked blue): we reverse-sort the packets list via 'No.' column – makes the last packet appear on top.

Method 3 (marked red): we look on the bottom of the window for the total amount of packets.

Question: What was the suspicious port that was being used?

Answer: 4444

Method: port 4444 is often used for reverse shell (often the default reverse shell port on meterpreter).


File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

tcp.stream eq 0

No.	Time	Source	Destination	Protocol	Length	Info
3	0.000215	10.129.43.29	10.129.43.4	TCP	66	50612 → 4444 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256…
4	0.000270	10.129.43.4	10.129.43.29	TCP	66	4444 → 50612 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS…
5	0.000415	10.129.43.29	10.129.43.4	TCP	60	50612 → 4444 [ACK] Seq=1 Ack=1 Win=2102272 Len=0
6	0.070797	10.129.43.29	10.129.43.4	TCP	175	50612 → 4444 [PSH, ACK] Seq=1 Ack=1 Win=2102272 Len=121
7	0.070843	10.129.43.4	10.129.43.29	TCP	54	4444 → 50612 [ACK] Seq=1 Ack=122 Win=64128 Len=0
8	10.676486	10.129.43.4	10.129.43.29	TCP	61	4444 → 50612 [PSH, ACK] Seq=1 Ack=122 Win=64128 Len=7
9	10.745086	10.129.43.29	10.129.43.4	TCP	60	50612 → 4444 [ACK] Seq=122 Ack=8 Win=2102272 Len=0
10	10.745121	10.129.43.29	10.129.43.4	TCP	110	50612 → 4444 [PSH, ACK] Seq=122 Ack=8 Win=2102272 Len=56
11	10.745135	10.129.43.4	10.129.43.29	TCP	54	4444 → 50612 [ACK] Seq=8 Ack=178 Win=64128 Len=0
12	15.202665	10.129.43.4	10.129.43.29	TCP	63	4444 → 50612 [PSH, ACK] Seq=8 Ack=178 Win=64128 Len=9
13	15.211515	10.129.43.29	10.129.43.4	TCP	64	50612 → 4444 [PSH, ACK] Seq=178 Ack=17 Win=2102272 Len=…

As can be seen on the screenshot – the TCP stream was conducted on port 4444.

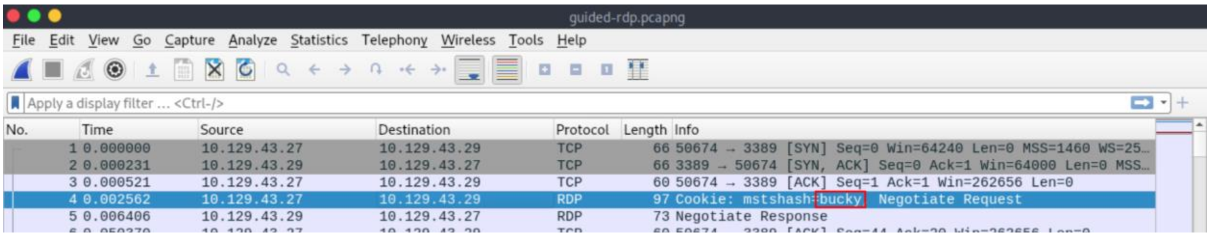
Decrypting RDP connections:

Question: What user account was used to initiate the RDP connection?

Answer: bucky

Method: First, lets download the 'RDP-Analysis-Resources' from the resources bag to the pwnbox.

Then - we unzip it, and open the extracted file 'guided-rdp.pcapng' with Wireshark.



guided-rdp.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>

```
No.   Time        Source         Destination    Protocol Length Info
    1 0.000000    10.129.43.27   10.129.43.29   TCP      66 50674 → 3389 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=25…
    2 0.000231    10.129.43.29   10.129.43.27   TCP      66 3389 → 50674 [SYN, ACK] Seq=0 Ack=1 Win=64000 Len=0 MSS…
    3 0.000521    10.129.43.27   10.129.43.29   TCP      60 50674 → 3389 [ACK] Seq=1 Ack=1 Win=262656 Len=0
    4 0.002562    10.129.43.27   10.129.43.29   RDP      97 Cookie: mstshash=bucky  Negotiate Request
    5 0.006406    10.129.43.29   10.129.43.27   RDP      73 Negotiate Response
```

Immediately upon opening the pcapng file – we can see the used account in the cookie field.