QUESTION 1

1 points Save Answer

From: coworker@workplace.com
To: you@workplace.com
Subject: Home System Problem

Hey - I'm having problems with my system connecting to Steam - and I can't play Elden Ring. I tried calling support from my ISP and they were asking me questions about my IPv4 and IPv6 settings (they said something about supporting both), but I didn't follow what they were saying. Can you look at my settings and tell me if the problem is that I don't have the right settings to connect to the Internet, or should I be contacting Steam support about problems with my account?

Let me know either way and, if you have anything I could say to my ISP or Steam to help them understand the problem, I would appreciate it.

Thanks.
Coworker

```
ens33     Link encap:Ethernet  HWaddr 00:0c:29:1d:0a:7b
          inet addr:10.0.0.155  Bcast:10.0.0.255  Mask:255.255.255.0
          inet6 addr: 2601:100:827e:1040:20c:29ff:fe1d:a7b/64 Scope:Global
          inet6 addr: fe80::20c:29ff:fe1d:a7b/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:60413 errors:0 dropped:0 overruns:0 frame:0
          TX packets:4701 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:87335914 (87.3 MB)  TX bytes:395673 (395.6 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:224 errors:0 dropped:0 overruns:0 frame:0
          TX packets:224 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:17664 (17.6 KB)  TX bytes:17664 (17.6 KB)

virbr0    Link encap:Ethernet  HWaddr 52:54:00:cc:bf:7f
          inet addr:192.168.122.1  Bcast:192.168.122.255  Mask:255.255.255.0
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

virbr0-nic Link encap:Ethernet  HWaddr 52:54:00:cc:bf:7f
          BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

QUESTION 2

1 points Save Answe

From: a.coworker@workplace.com
To: you@workplace.com
Subject: Jump Box

Hey – I am setting up a jump box so people on LAN_A can ssh into it and then ssh out to a host on LAN_B. Or go from LAN_B to LAN_A. (The other direction.)
I don't have credentials for hosts on those subnets since LAN_A is for the financial record managers and LAN_B the health record managers – and I'm not in either of those groups – so I can't test that it works. I would talk to those folks and have them test it for me – but it is already after 6:00 P.M. on a Friday (so everyone is gone) and I just got asked if I have it done by the_boss.
Can you look at the two attached images and tell me what interfaces the SSH Daemon service is listening on? (Or not listening on?) Also, can you give me the breakdown by IPv4 and IPv6 – you know the_boss loves being specific about both protocols.
I will owe you one!

a.coworker

```
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 192.168.122.1:53        0.0.0.0:*               LISTEN      2082/dnsmasq
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      1320/sshd
tcp6       0      0 :::22                   :::*                    LISTEN      1320/sshd
udp        0      0 192.168.122.1:53        0.0.0.0:*                           2082/dnsmasq
udp        0      0 0.0.0.0:67              0.0.0.0:*                           2082/dnsmasq
udp        0      0 0.0.0.0:68              0.0.0.0:*                           1155/dhclient
```

```
ens33     Link encap:Ethernet  HWaddr 00:0c:29:1d:0a:7b
          inet addr:10.0.0.155  Bcast:10.0.0.255  Mask:255.255.255.0
          inet6 addr: 2601:100:827e:1040:20c:29ff:fe1d:a7b/64 Scope:Global
          inet6 addr: fe80::20c:29ff:fe1d:a7b/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:60413 errors:0 dropped:0 overruns:0 frame:0
          TX packets:4701 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:87335914 (87.3 MB)  TX bytes:395673 (395.6 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:224 errors:0 dropped:0 overruns:0 frame:0
          TX packets:224 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:17664 (17.6 KB)  TX bytes:17664 (17.6 KB)

virbr0    Link encap:Ethernet  HWaddr 52:54:00:cc:bf:7f
          inet addr:192.168.122.1  Bcast:192.168.122.255  Mask:255.255.255.0
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

virbr0-nic Link encap:Ethernet  HWaddr 52:54:00:cc:bf:7f
          BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

QUESTION 3

1 points Save Ans

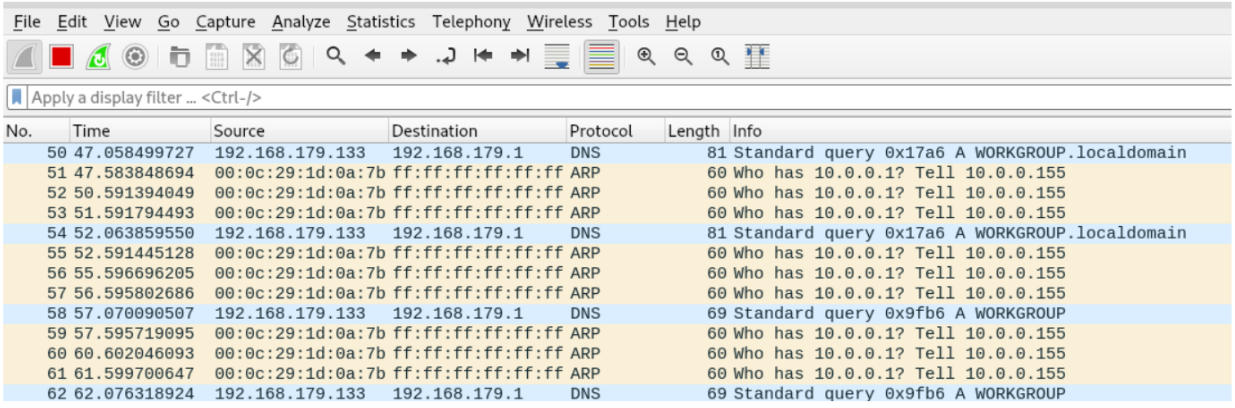
This question is directly testing your understanding of the network fundamentals lesson.

This Wireshark capture below shows the host is receiving a number of ARP broadcast messages. (The destination address being **ff:ff:ff:ff:ff:ff** indicates it is a broadcast message.) The information Wireshark is showing also tells the user that it is 10.0.0.155 that is asking for the information.

However, the second image shows that when the user tries to ping 10.0.0.155 they get the error: **Network is unreachable.**

At first glance, this seems weird and something must be wrong. So, the user runs the "ifconfig -a" command to figure out what the problem might be. This gives them the information on why the pings aren't working.

The two pieces of information to answer this question are: What did they see in the command output that told them what the problem was (and what was the problem), and why can they see the ARP traffic in the first place?



No.	Time	Source	Destination	Protocol	Length	Info
50	47.058499727	192.168.179.133	192.168.179.1	DNS	81	Standard query 0x17a6 A WORKGROUP.localdomain
51	47.583848694	00:0c:29:1d:0a:7b	ff:ff:ff:ff:ff:ff	ARP	60	Who has 10.0.0.1? Tell 10.0.0.155
52	50.591394049	00:0c:29:1d:0a:7b	ff:ff:ff:ff:ff:ff	ARP	60	Who has 10.0.0.1? Tell 10.0.0.155
53	51.591794493	00:0c:29:1d:0a:7b	ff:ff:ff:ff:ff:ff	ARP	60	Who has 10.0.0.1? Tell 10.0.0.155
54	52.063859550	192.168.179.133	192.168.179.1	DNS	81	Standard query 0x17a6 A WORKGROUP.localdomain
55	52.591445128	00:0c:29:1d:0a:7b	ff:ff:ff:ff:ff:ff	ARP	60	Who has 10.0.0.1? Tell 10.0.0.155
56	55.596696205	00:0c:29:1d:0a:7b	ff:ff:ff:ff:ff:ff	ARP	60	Who has 10.0.0.1? Tell 10.0.0.155
57	56.595802686	00:0c:29:1d:0a:7b	ff:ff:ff:ff:ff:ff	ARP	60	Who has 10.0.0.1? Tell 10.0.0.155
58	57.070090507	192.168.179.133	192.168.179.1	DNS	69	Standard query 0x9fb6 A WORKGROUP
59	57.595719095	00:0c:29:1d:0a:7b	ff:ff:ff:ff:ff:ff	ARP	60	Who has 10.0.0.1? Tell 10.0.0.155
60	60.602046093	00:0c:29:1d:0a:7b	ff:ff:ff:ff:ff:ff	ARP	60	Who has 10.0.0.1? Tell 10.0.0.155
61	61.599700647	00:0c:29:1d:0a:7b	ff:ff:ff:ff:ff:ff	ARP	60	Who has 10.0.0.1? Tell 10.0.0.155
62	62.076318924	192.168.179.133	192.168.179.1	DNS	69	Standard query 0x9fb6 A WORKGROUP

```
root@kali:~# ping 10.0.0.155
ping: connect: Network is unreachable
root@kali:~# ping 10.0.0.155
ping: connect: Network is unreachable
root@kali:~# ifconfig -a
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.179.133  netmask 255.255.255.0  broadcast 192.168.179.255
        ether 00:0c:29:39:46:7e  txqueuelen 1000  (Ethernet)
        RX packets 1064  bytes 143157 (139.8 KiB)
        RX errors 0  dropped 64  overruns 0  frame 0
        TX packets 215  bytes 19927 (19.4 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth1: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        ether 00:50:56:3a:ee:1b  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 48  bytes 2512 (2.4 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 48  bytes 2512 (2.4 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

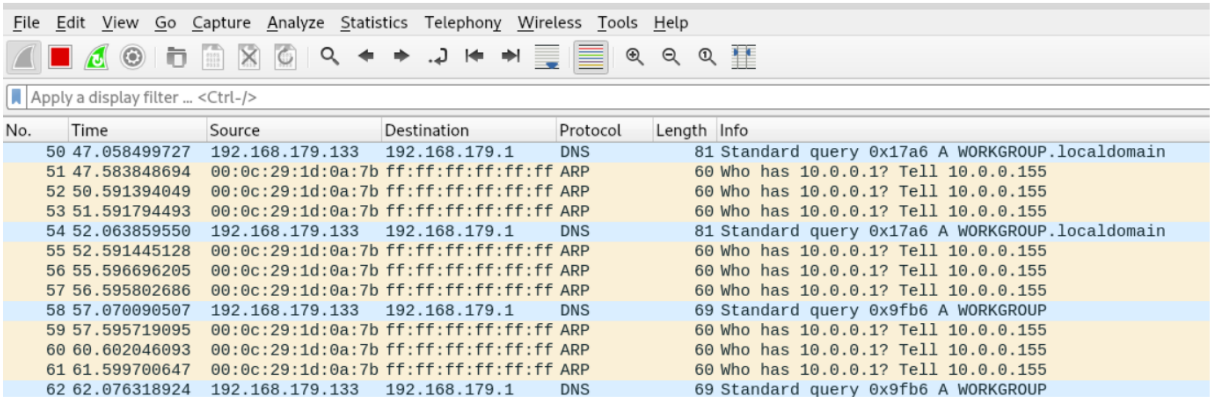
QUESTION 4

1 points Save Answe

This question digs deeper into the information that Wireshark gives us. (Which is always a lot!)

We can see in the "Packet List Pane" – in the "Info" area of the lines highlighted in yellow that Wireshark is telling us the ARP request is "Who has 10.0.0.1? Tell 10.0.0.155"

Given the information in the "Packet Details Pane" (the section that starts with "`> Frame 52:`") how can you validate the IP address of which host sent the packet – and how do you know the IP address that it is requesting the MAC address for? (i.e. – How does Wireshark know that the question is: "Who has 10.0.0.1? Tell 10.0.0.155"



No.	Time	Source	Destination	Protocol	Length	Info
50	47.058499727	192.168.179.133	192.168.179.1	DNS	81	Standard query 0x17a6 A WORKGROUP.localdomain
51	47.583848694	00:0c:29:1d:0a:7b	ff:ff:ff:ff:ff:ff	ARP	60	Who has 10.0.0.1? Tell 10.0.0.155
52	50.591394049	00:0c:29:1d:0a:7b	ff:ff:ff:ff:ff:ff	ARP	60	Who has 10.0.0.1? Tell 10.0.0.155
53	51.591794493	00:0c:29:1d:0a:7b	ff:ff:ff:ff:ff:ff	ARP	60	Who has 10.0.0.1? Tell 10.0.0.155
54	52.063859550	192.168.179.133	192.168.179.1	DNS	81	Standard query 0x17a6 A WORKGROUP.localdomain
55	52.591445128	00:0c:29:1d:0a:7b	ff:ff:ff:ff:ff:ff	ARP	60	Who has 10.0.0.1? Tell 10.0.0.155
56	55.596696205	00:0c:29:1d:0a:7b	ff:ff:ff:ff:ff:ff	ARP	60	Who has 10.0.0.1? Tell 10.0.0.155
57	56.595802686	00:0c:29:1d:0a:7b	ff:ff:ff:ff:ff:ff	ARP	60	Who has 10.0.0.1? Tell 10.0.0.155
58	57.070090507	192.168.179.133	192.168.179.1	DNS	69	Standard query 0x9fb6 A WORKGROUP
59	57.595719095	00:0c:29:1d:0a:7b	ff:ff:ff:ff:ff:ff	ARP	60	Who has 10.0.0.1? Tell 10.0.0.155
60	60.602046093	00:0c:29:1d:0a:7b	ff:ff:ff:ff:ff:ff	ARP	60	Who has 10.0.0.1? Tell 10.0.0.155
61	61.599700647	00:0c:29:1d:0a:7b	ff:ff:ff:ff:ff:ff	ARP	60	Who has 10.0.0.1? Tell 10.0.0.155
62	62.076318924	192.168.179.133	192.168.179.1	DNS	69	Standard query 0x9fb6 A WORKGROUP

Wireshark · Packet 52 · eth0

```
Frame 52: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0
Ethernet II, Src: 00:0c:29:1d:0a:7b, Dst: ff:ff:ff:ff:ff:ff
  Destination: ff:ff:ff:ff:ff:ff
     Address: ff:ff:ff:ff:ff:ff
     .... ..1. .... .... .... .... = LG bit: Locally administered address (this is NOT the
     .... ...1 .... .... .... .... = IG bit: Group address (multicast/broadcast)
  Source: 00:0c:29:1d:0a:7b
     Address: 00:0c:29:1d:0a:7b
     .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
     .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Type: ARP (0x0806)
    Padding: 000000000000000000000000000000000000
Address Resolution Protocol (request)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: request (1)
    Sender MAC address: 00:0c:29:1d:0a:7b
    Sender IP address: 10.0.0.155
    Target MAC address: 00:00:00:00:00:00
    Target IP address: 10.0.0.1
```

```
0000  ff ff ff ff ff ff 00 0c  29 1d 0a 7b 08 06 00 01   ........ )..{....
0010  08 00 06 04 00 01 00 0c  29 1d 0a 7b 0a 00 00 9b   ........ )..{....
0020  00 00 00 00 00 00 0a 00  00 01 00 00 00 00 00 00   ........ ........
0030  00 00 00 00 00 00 00 00  00 00 00 00               ........ ....
```

QUESTION 5

1 points Save Answer

Same scenario – but digging into the "Packet Bytes Pane" of Wireshark.

If we didn't have Wireshark installed on a host that we had compromised, we might have to use something with far fewer features – like the tcpdump command, or some other really lightweight tool. This kind of tool may only be able to give us the raw output – without any kind of analysis to help us – which would essentially be just like the information we see in the Packet Bytes Pane. (Which starts with 0000 and a lot of ff ff ff…)

We know the IP address of the host sending the ARP packet is 10.0.0.155, and we know it is in that list of bytes somewhere. So, using "Base 10 notation" at what "byte offset" can we find that IP address?

A few things to keep in mind…

Each row starts at an offset of "0" (zero) so the ***offset*** of the first "29" in the top row is 8, even though it is the 9th value. So, watch out for the dreaded "off by one" error. Also, keep in mind that the display is showing you the information in hexadecimal and not "Base 10" so you aren't going to see "155" displayed in there, you will have to convert it to hexadecimal first. (***Note***: Programmer mode in the Windows calculator can handle that conversion.)

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter … <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
50	47.058499727	192.168.179.133	192.168.179.1	DNS	81	Standard query 0x17a6 A WORKGROUP.localdomain
51	47.583848694	00:0c:29:1d:0a:7b	ff:ff:ff:ff:ff:ff	ARP	60	Who has 10.0.0.1? Tell 10.0.0.155
52	50.591394049	00:0c:29:1d:0a:7b	ff:ff:ff:ff:ff:ff	ARP	60	Who has 10.0.0.1? Tell 10.0.0.155
53	51.591794493	00:0c:29:1d:0a:7b	ff:ff:ff:ff:ff:ff	ARP	60	Who has 10.0.0.1? Tell 10.0.0.155
54	52.063859550	192.168.179.133	192.168.179.1	DNS	81	Standard query 0x17a6 A WORKGROUP.localdomain
55	52.591445128	00:0c:29:1d:0a:7b	ff:ff:ff:ff:ff:ff	ARP	60	Who has 10.0.0.1? Tell 10.0.0.155
56	55.596696205	00:0c:29:1d:0a:7b	ff:ff:ff:ff:ff:ff	ARP	60	Who has 10.0.0.1? Tell 10.0.0.155
57	56.595802686	00:0c:29:1d:0a:7b	ff:ff:ff:ff:ff:ff	ARP	60	Who has 10.0.0.1? Tell 10.0.0.155
58	57.070090507	192.168.179.133	192.168.179.1	DNS	69	Standard query 0x9fb6 A WORKGROUP
59	57.595719095	00:0c:29:1d:0a:7b	ff:ff:ff:ff:ff:ff	ARP	60	Who has 10.0.0.1? Tell 10.0.0.155
60	60.602046093	00:0c:29:1d:0a:7b	ff:ff:ff:ff:ff:ff	ARP	60	Who has 10.0.0.1? Tell 10.0.0.155
61	61.599700647	00:0c:29:1d:0a:7b	ff:ff:ff:ff:ff:ff	ARP	60	Who has 10.0.0.1? Tell 10.0.0.155
62	62.076318924	192.168.179.133	192.168.179.1	DNS	69	Standard query 0x9fb6 A WORKGROUP

Wireshark · Packet 52 · eth0

```
▸ Frame 52: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0
▾ Ethernet II, Src: 00:0c:29:1d:0a:7b, Dst: ff:ff:ff:ff:ff:ff
  ▾ Destination: ff:ff:ff:ff:ff:ff
      Address: ff:ff:ff:ff:ff:ff
      .... ..1. .... .... .... .... = LG bit: Locally administered address (this is NOT the
      .... ...1 .... .... .... .... = IG bit: Group address (multicast/broadcast)
  ▾ Source: 00:0c:29:1d:0a:7b
      Address: 00:0c:29:1d:0a:7b
      .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
      .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Type: ARP (0x0806)
    Padding: 000000000000000000000000000000000000
▾ Address Resolution Protocol (request)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: request (1)
    Sender MAC address: 00:0c:29:1d:0a:7b
    Sender IP address: 10.0.0.155
    Target MAC address: 00:00:00:00:00:00
    Target IP address: 10.0.0.1
```

```
0000  ff ff ff ff ff ff 00 0c  29 1d 0a 7b 08 06 00 01   ........ )..{....
0010  08 00 06 04 00 01 00 0c  29 1d 0a 7b 0a 00 00 9b   ........ )..{....
0020  00 00 00 00 00 00 0a 00  00 01 00 00 00 00 00 00   ........ ........
0030  00 00 00 00 00 00 00 00  00 00 00 00               ........ ....
```

QUESTION 6

This is Part I of a two-part challenge from your professor and is captured in this question and the next question.
The screenshot below shows the ICMP ECHO REQUEST and the ICMP ECHO REPLY from two different nodes. From the information provided:

a. What is the IP address of the host most likely running a version of ***Windows***?

b. How did you determine which one it was?

```
root@kali:~# ping -c2 -W3 -vv 10.0.0.121
PING 10.0.0.121 (10.0.0.121) 56(84) bytes of data.
64 bytes from 10.0.0.121: icmp_seq=1 ttl=128 time=1.22 ms
64 bytes from 10.0.0.121: icmp_seq=2 ttl=128 time=0.393 ms

--- 10.0.0.121 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1003ms
rtt min/avg/max/mdev = 0.393/0.808/1.224/0.415 ms
root@kali:~# ping -c2 -W3 -vv 10.0.0.155
PING 10.0.0.155 (10.0.0.155) 56(84) bytes of data.
64 bytes from 10.0.0.155: icmp_seq=1 ttl=64 time=1.21 ms
64 bytes from 10.0.0.155: icmp_seq=2 ttl=64 time=0.455 ms

--- 10.0.0.155 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 0.455/0.833/1.211/0.378 ms
```

QUESTION 7

This is Part II of a two-part challenge from your professor and is captured in this question and the previous question.
The screenshot below shows the ICMP ECHO REQUEST and the ICMP ECHO REPLY from two different nodes. From the information provided:

a. What is the IP address of the host most likely running a version of ***Linux***?

b. How did you determine which one it was?

```
root@kali:~# ping -c2 -W3 -vv 10.0.0.121
PING 10.0.0.121 (10.0.0.121) 56(84) bytes of data.
64 bytes from 10.0.0.121: icmp_seq=1 ttl=128 time=1.22 ms
64 bytes from 10.0.0.121: icmp_seq=2 ttl=128 time=0.393 ms

--- 10.0.0.121 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1003ms
rtt min/avg/max/mdev = 0.393/0.808/1.224/0.415 ms
root@kali:~# ping -c2 -W3 -vv 10.0.0.155
PING 10.0.0.155 (10.0.0.155) 56(84) bytes of data.
64 bytes from 10.0.0.155: icmp_seq=1 ttl=64 time=1.21 ms
64 bytes from 10.0.0.155: icmp_seq=2 ttl=64 time=0.455 ms

--- 10.0.0.155 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 0.455/0.833/1.211/0.378 ms
```

QUESTION 8

1 points Save Answ

From: coworker@workplace.com
To: you@workplace.com
Subject: Vendor Training Class

Hey – so I am at the vendor training for the company that is supplying our new end point monitoring solution and ran into a problem. They have a cool library of scripts that can pull data from end systems and summarize how many systems are running things like sshd or a web server so you can get a summary of everything running on your network. So you could know that you have 135 web servers running on 108 different hosts.

Their script will handle the parsing of the information on each system and sending it to the central repository where it can be queried, summarized, and generated reports. However, you still have to run the right commands to generate the information they need.

I ended up getting to training a bit late this morning, so I missed the list of commands (and it isn't in the handouts.) I do have an image of the command they used to create the input for their scripts, but it was just a graphic and the actual command itself was cut off.

Since you are the expert when it comes to network stuff, I figured you would be familiar with the commands to see the state of your network connections. So, do you know what command and flag combination produces the output that is shown below?

Oh, and the instructor said you didn't want to see "ESTABLISHED" connections since each new one would show up as a separate record in the database. (Or something like that – I was still working on my first coffee.) Oh, and also that it does need to be exact or their scripts will parse and save it differently.

Anyway, anything you can tell me would be helpful.

Thanks.
Coworker

```
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 192.168.122.1:53        0.0.0.0:*               LISTEN      2082/dnsmasq
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      1320/sshd
tcp6       0      0 :::22                   :::*                    LISTEN      1320/sshd
udp        0      0 192.168.122.1:53        0.0.0.0:*                           2082/dnsmasq
udp        0      0 0.0.0.0:67              0.0.0.0:*                           2082/dnsmasq
udp        0      0 0.0.0.0:68              0.0.0.0:*                           1155/dhclient
```

QUESTION 9

1 points Save Answe

From: coworker@workplace.com
To: you@workplace.com
Subject: Vendor Training Class

Sorry. One more command…The instructor also said that the script expects to see the "Skipped" and "Found" so it can summarize those as well. Also, the phrase "expected to be exactly like this" was said I don't know how many times during the Data Per Interface class.

You are awesome!

Thanks
Coworker

> *>From: coworker@workplace.com*
> *>To: you@workplace.com*
> *>Subject: Vendor Training Class*
>
> *>Hey – so I am at the vendor training for the company that is supplying our new end point monitoring*
> *>solution and ran into a problem. They have a cool library of scripts that can pull data from end systems*
> *>and summarize how many systems are running things like sshd or a web server so you can get a*
> *>summary of everything running on your network. So you could know that you have 135 web servers*
> *>running on 108 different hosts.*
>
> *>Their script will handle the parsing of the information on each system and sending it to the central*
> *…*

```
Address                  HWtype  HWaddress           Flags Mask            Iface
10.0.0.1                 ether   bc:2e:48:8d:61:e0   C                     ens33
10.0.0.138               ether   00:0c:29:39:46:7e   C                     ens33
10.0.0.125                       (incomplete)                              ens33
10.0.0.175               ether   00:0c:29:b3:96:03   C                     ens33
10.0.0.139               ether   e4:ce:8f:5e:cd:a3   C                     ens33
Entries: 5      Skipped: 0      Found: 5
```

QUESTION 10

1 points Save Answe

From: fellow.student@my.utsa.edu
To: you@ my.utsa.edu
Subject: IS-4543 Lab Environment

So, I was working on my lab and the cat knocked over my coffee and all over my notes with my VM information – so I lost their IP addresses, MAC addresses, and everything else. Normally, I would just pull that up in VMware Workstation, or the ESXi interface, but I am currently tunneled into my environment with an SSH connection since I'm on the coast for a few days.

Anyway, I need to figure out which of the following IP addresses belong to my VMs so I can remotely reboot them. However, if I accidentally reboot one of my physical hosts, I'll lose my tunnel, and I will have to head home early to finish up the lab – and I like the beach!

I remember the professor said something about being able to figure out whether something was a VM or not with the information below, so I was wondering if you could help me out. Also, please let me know HOW you know what's a VM (or not) so I can handle it myself if I run into this situation again.

Thanks.
Fellow Student

```
Address                  HWtype  HWaddress           Flags Mask            Iface
10.0.0.1                 ether   bc:2e:48:8d:61:e0   C                     ens33
10.0.0.138               ether   00:0c:29:39:46:7e   C                     ens33
10.0.0.125                       (incomplete)                              ens33
10.0.0.175               ether   00:0c:29:b3:96:03   C                     ens33
10.0.0.139               ether   e4:ce:8f:5e:cd:a3   C                     ens33
Entries: 5      Skipped: 0      Found: 5
```