10.2.7 Lab - Using Wireshark to Examine a UDP DNS Capture

This lab has been updated for use on NETLAB+.
www.netdevgroup.com

Objectives

Part 1: Record a PC's IP Configuration Information

Part 2: Use Wireshark to Capture DNS Queries and Responses

Part 3: Analyze Captured DNS or UDP Packets

Background / Scenario

When you use the internet, you use the Domain Name System (DNS). DNS is a distributed network of servers that translates user-friendly domain names like www.google.com to an IP address. When you type a website URL into your browser, your PC performs a DNS query to the DNS server's IP address. Your PC's DNS query and the DNS server's response make use of the User Datagram Protocol (UDP) as the transport layer protocol. UDP is connectionless and does not require a session setup as does TCP. DNS queries and responses are very small and do not require the overhead of TCP.

In this lab, you will communicate with a DNS server by sending a DNS query using the UDP transport protocol. You will use Wireshark to examine the DNS query and response exchanges with the same server.

Instructions

Part 1: Record VM's IP Configuration Information

In *Part 1*, you will use commands on the *Workstation* VM to find and record the MAC and IP addresses of your VM's virtual network interface card (NIC), the IP address of the specified default gateway, and the DNS server IP address specified for the PC. Record this information in the table provided. The information will be used in parts of this lab with packet analysis.

Description	Settings
IP address	
MAC address	
Default gateway IP address	
DNS server IP address	

a. Launch the **Workstation** VM. Log in with username `analyst` and the password `cyberops`.

Open a terminal in the VM. Enter **ifconfig** at the prompt to display interface information.

Note: In Part 1, your results will vary depending on your local area network settings and internet connection.

```
[analyst@secOps ~]$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.8.10  netmask 255.255.255.0  broadcast 192.168.8.255
        inet6 fe80::a00:27ff:fe82:75df  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:82:75:df  txqueuelen 1000  (Ethernet)
        RX packets 41953  bytes 14354223 (13.6 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 15249  bytes 1723493 (1.6 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
<some output omitted>
```

b. At the terminal prompt, enter **cat /etc/resolv.conf** to determine the DNS server.

```
[analyst@secOps ~]$ cat /etc/resolv.conf
# Resolver configuration file.
# See resolv.conf(5) for details.
nameserver 8.8.4.4
nameserver 209.165.200.235
```

c. At the terminal prompt, enter **netstat -rn** to display the IP routing table to the default gateway IP address.

```
[analyst@secOps ~]$ netstat -rn
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
0.0.0.0         192.168.8.1     0.0.0.0         UG        0 0          0 enp0s3
192.168.8.0     0.0.0.0         255.255.255.0   U         0 0          0 enp0s3
192.168.8.1     0.0.0.0         255.255.255.255 UH        0 0          0 enp0s3
```

Note: The DNS IP address and default gateway IP address are often the same, especially in small networks. However, in a business or school network, the addresses would most likely be different.

Part 2: Analyze Captured DNS or UDP Packets

In Part 3, you will examine the UDP packets that were generated when communicating with a DNS server for the IP addresses for www.google.com.

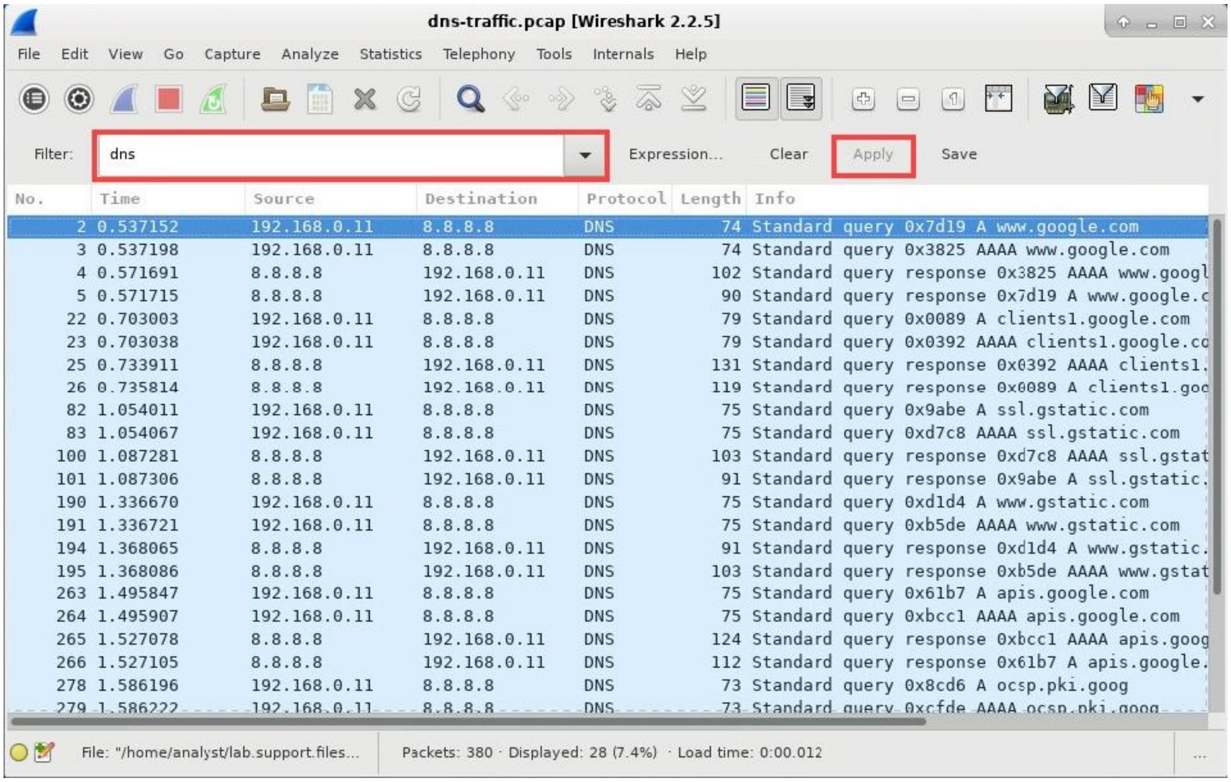
Step 1: Filter DNS packets.

a. In the terminal window, start Wireshark, enter `cyberops` as password and click **OK** when prompted.

```
[analyst@secOps ~]$ sudo wireshark
[sudo] password for analyst:
```

b. In the *Wireshark* main window, navigate to **File > Open** and open the **dns-traffic.pcap** file from *~/lab.support.files/pcaps/*.

c. Filter the results by typing `dns` in the *Filter* field. Click **Apply** (arrow) or press the **Enter** key to apply the filter**.**


dns-traffic.pcap [Wireshark 2.2.5]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Expression... Clear Apply Save

Filter: dns

No.	Time	Source	Destination	Protocol	Length	Info
2	0.537152	192.168.0.11	8.8.8.8	DNS	74	Standard query 0x7d19 A www.google.com
3	0.537198	192.168.0.11	8.8.8.8	DNS	74	Standard query 0x3825 AAAA www.google.com
4	0.571691	8.8.8.8	192.168.0.11	DNS	102	Standard query response 0x3825 AAAA www.googl
5	0.571715	8.8.8.8	192.168.0.11	DNS	90	Standard query response 0x7d19 A www.google.c
22	0.703003	192.168.0.11	8.8.8.8	DNS	79	Standard query 0x0089 A clients1.google.com
23	0.703038	192.168.0.11	8.8.8.8	DNS	79	Standard query 0x0392 AAAA clients1.google.co
25	0.733911	8.8.8.8	192.168.0.11	DNS	131	Standard query response 0x0392 AAAA clients1.
26	0.735814	8.8.8.8	192.168.0.11	DNS	119	Standard query response 0x0089 A clients1.goo
82	1.054011	192.168.0.11	8.8.8.8	DNS	75	Standard query 0x9abe A ssl.gstatic.com
83	1.054067	192.168.0.11	8.8.8.8	DNS	75	Standard query 0xd7c8 AAAA ssl.gstatic.com
100	1.087281	8.8.8.8	192.168.0.11	DNS	103	Standard query response 0xd7c8 AAAA ssl.gstat
101	1.087306	8.8.8.8	192.168.0.11	DNS	91	Standard query response 0x9abe A ssl.gstatic.
190	1.336670	192.168.0.11	8.8.8.8	DNS	75	Standard query 0xd1d4 A www.gstatic.com
191	1.336721	192.168.0.11	8.8.8.8	DNS	75	Standard query 0xb5de AAAA www.gstatic.com
194	1.368065	8.8.8.8	192.168.0.11	DNS	91	Standard query response 0xd1d4 A www.gstatic.
195	1.368086	8.8.8.8	192.168.0.11	DNS	103	Standard query response 0xb5de AAAA www.gstat
263	1.495847	192.168.0.11	8.8.8.8	DNS	75	Standard query 0x61b7 A apis.google.com
264	1.495907	192.168.0.11	8.8.8.8	DNS	75	Standard query 0xbcc1 AAAA apis.google.com
265	1.527078	8.8.8.8	192.168.0.11	DNS	124	Standard query response 0xbcc1 AAAA apis.goog
266	1.527105	8.8.8.8	192.168.0.11	DNS	112	Standard query response 0x61b7 A apis.google.
278	1.586196	192.168.0.11	8.8.8.8	DNS	73	Standard query 0x8cd6 A ocsp.pki.goog
279	1.586222	192.168.0.11	8.8.8.8	DNS	73	Standard query 0xcfde AAAA ocsp.pki.goog

File: "/home/analyst/lab.support.files... Packets: 380 · Displayed: 28 (7.4%) · Load time: 0:00.012

d. In the packet list pane (top section) of the main window, locate the packet that includes *Standard query* and *A www.google.com*. See frame 2 above as an example.

Step 2: Examine the fields in a DNS query packet.

The protocol fields, highlighted in gray, are displayed in the packet details pane (middle section) of the main window.

a. In the first line in the packet details pane, frame 2 had 74 bytes of data on the wire. This is the number of bytes it took to send a DNS query to a named server requesting the IP addresses of www.google.com. If you used a different web address, such as www.cisco.com, the byte count might be different.

b. The Ethernet II line displays the source and destination MAC addresses. The source MAC address is from your VM because your VM originated the DNS query. The destination MAC address is from the default gateway because this is the last stop before this query exits the local network.

Is the source MAC address the same as the one recorded from Part 1 for the VM?

____

c. In the Internet Protocol Version 4 line, the IP packet Wireshark capture indicates that the source IP address of this DNS query is 192.168.0.11 and the destination IP address is 8.8.8.8. In this example, the destination address is the DNS server.

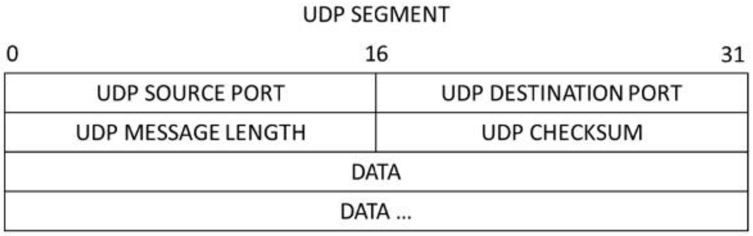
Can you identify the IP and MAC addresses for the source and destinations of this packet?

Device	IP Address	MAC Address
Source Workstation		
Destination DNS Server/ Default Gateway		

Note: The destination IP address is for the DNS Server, but the destination MAC address is for the default gateway.

The IP packet and header encapsulates the UDP segment. The UDP segment contains the DNS query as the data.

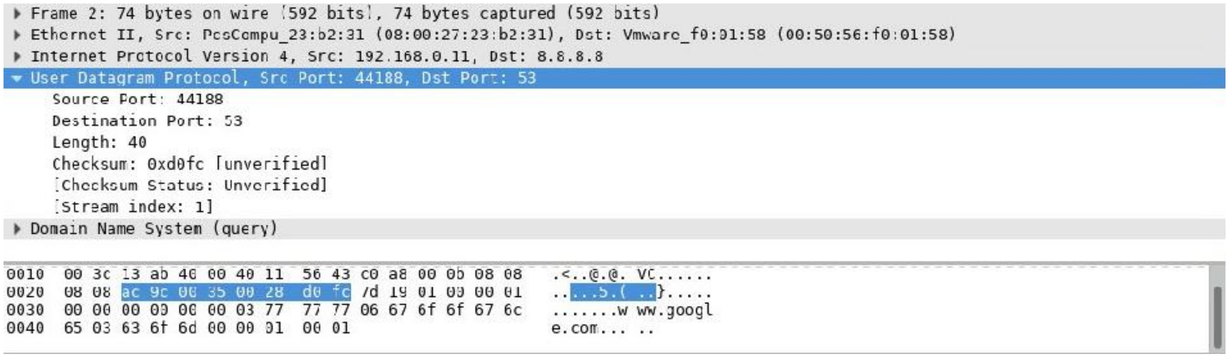
d. A UDP header only has four fields: source port, destination port, length, and checksum. Each field in a UDP header is only 16 bits as depicted below.



UDP SEGMENT

0	16	31
UDP SOURCE PORT	UDP DESTINATION PORT	
UDP MESSAGE LENGTH	UDP CHECKSUM	
DATA		
DATA ...		

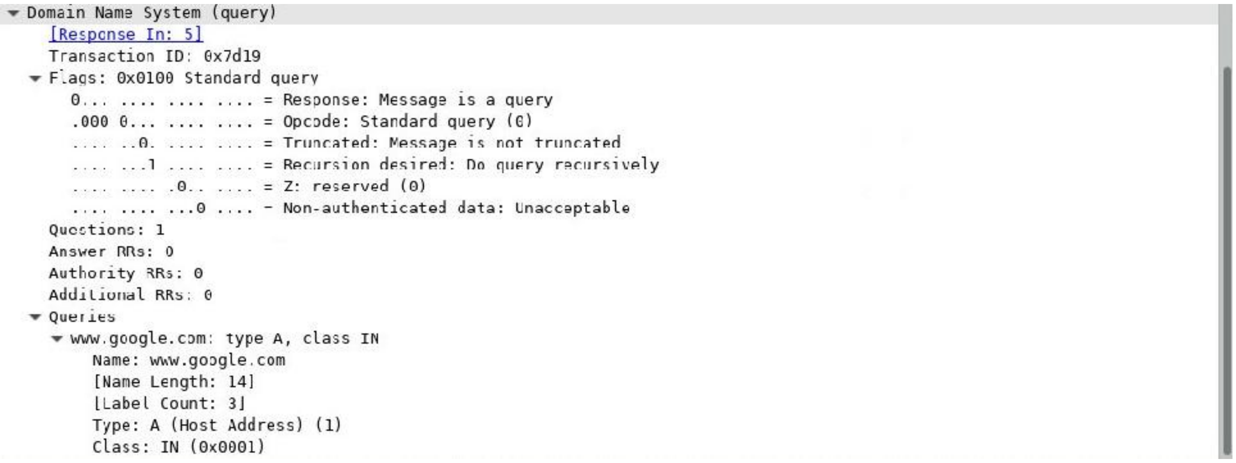
Click the arrow next to **User Datagram Protocol** to view the details. Notice that there are only four fields. The source port number in this example is 58029. The source port was randomly generated by the VM using port numbers that are not reserved. The destination port is 53. Port 53 is a well-known port reserved for use with DNS. DNS servers listen on port 53 for DNS queries from clients.



```
▸ Frame 2: 74 bytes on wire (592 bits), 74 bytes captured (592 bits)
▸ Ethernet II, Src: PcsCompu_23:b2:31 (08:00:27:23:b2:31), Dst: Vmware_f0:01:58 (00:50:56:f0:01:58)
▸ Internet Protocol Version 4, Src: 192.168.0.11, Dst: 8.8.8.8
▾ User Datagram Protocol, Src Port: 44188, Dst Port: 53
    Source Port: 44188
    Destination Port: 53
    Length: 40
    Checksum: 0xd0fc [unverified]
    [Checksum Status: Unverified]
    [Stream index: 1]
▸ Domain Name System (query)

0010  00 3c 13 ab 40 00 40 11  56 43 c0 a8 00 0b 08 08   .<..@.@. VC......
0020  08 08 ac 9c 00 35 00 28  d0 fc 7d 19 01 00 00 01   .....5.( ..}.....
0030  00 00 00 00 00 00 03 77  77 77 06 67 6f 6f 67 6c   .......w ww.googl
0040  65 03 63 6f 6d 00 00 01  00 01                     e.com... ..
```

In this example, the length of the UDP segment is 40 bytes. The length of the UDP segment in your example may be different. Out of 40 bytes, 8 bytes are used as the header. The other 32 bytes are used by DNS query data. The 32 bytes of DNS query data is in the following illustration in the packet bytes pane (lower section) of the *Wireshark* main window.



```
▾ Domain Name System (query)
    [Response In: 5]
    Transaction ID: 0x7d19
  ▾ Flags: 0x0100 Standard query
      0... .... .... .... = Response: Message is a query
      .000 0... .... .... = Opcode: Standard query (0)
      .... ..0. .... .... = Truncated: Message is not truncated
      .... ...1 .... .... = Recursion desired: Do query recursively
      .... .... .0.. .... = Z: reserved (0)
      .... .... ...0 .... = Non-authenticated data: Unacceptable
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 0
  ▾ Queries
    ▾ www.google.com: type A, class IN
        Name: www.google.com
        [Name Length: 14]
        [Label Count: 3]
        Type: A (Host Address) (1)
        Class: IN (0x0001)
```

The checksum is used to determine the integrity of the UDP header after it has traversed the internet.

The UDP header has low overhead because UDP does not have fields that are associated with the three-way handshake in TCP. Any data transfer reliability issues that occur must be handled by the application layer.

Expand as necessary to see the details. Record your Wireshark results in the table below:

Description	Wireshark Results
Frame size	
Source MAC address	
Destination MAC address	
Source IP address	
Destination IP address	
Source port	
Destination port	

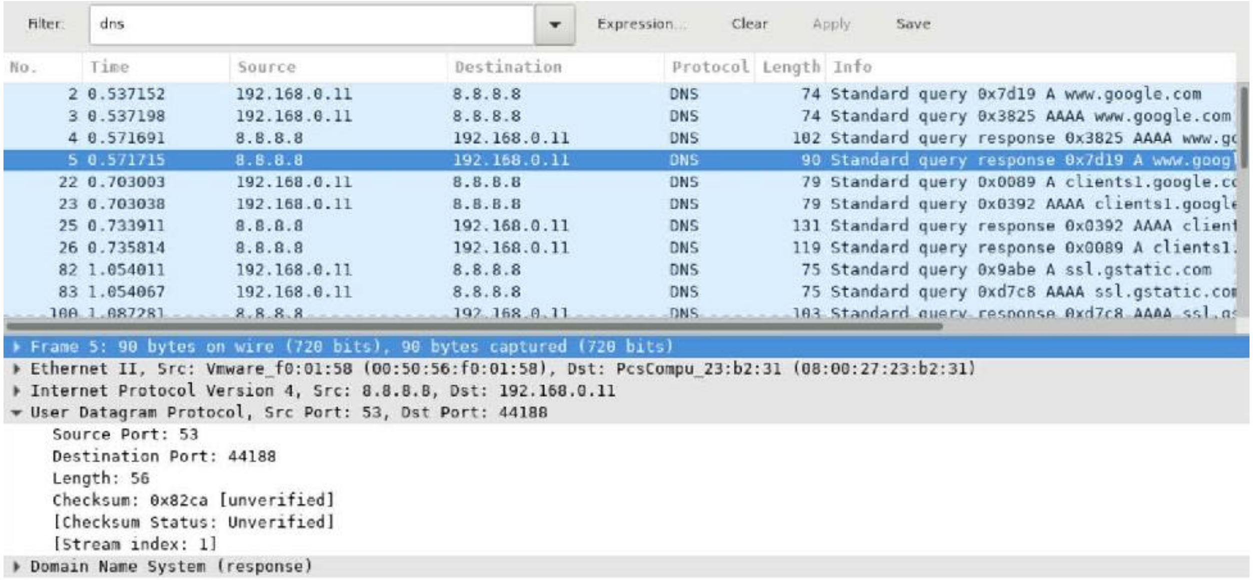
Is the source IP address the same as the local PC's IP address you recorded in Part 1?

Is the destination IP address the same as the default gateway noted in Part 1?

Step 3: Examine the fields in a DNS response packet.

In this step, you will examine the DNS response packet and verify that the DNS response packet also uses the UDP.

a. In this example, frame 5 is the corresponding DNS response packet. Notice the number of bytes on the wire is 102. It is a larger packet compared to the DNS query packet. This is because the DNS response packet will include a variety of information about the domain.



```
Filter: dns

No.  Time      Source        Destination   Protocol Length Info
  2 0.537152  192.168.0.11  8.8.8.8       DNS          74 Standard query 0x7d19 A www.google.com
  3 0.537198  192.168.0.11  8.8.8.8       DNS          74 Standard query 0x3825 AAAA www.google.com
  4 0.571691  8.8.8.8       192.168.0.11  DNS         102 Standard query response 0x3825 AAAA www.gc
  5 0.571715  8.8.8.8       192.168.0.11  DNS          90 Standard query response 0x7d19 A www.googl
 22 0.703003  192.168.0.11  8.8.8.8       DNS          79 Standard query 0x0089 A clients1.google.cc
 23 0.703038  192.168.0.11  8.8.8.8       DNS          79 Standard query 0x0392 AAAA clients1.google
 25 0.733911  8.8.8.8       192.168.0.11  DNS         131 Standard query response 0x0392 AAAA client
 26 0.735814  8.8.8.8       192.168.0.11  DNS         119 Standard query response 0x0089 A clients1.
 82 1.054011  192.168.0.11  8.8.8.8       DNS          75 Standard query 0x9abe A ssl.gstatic.com
 83 1.054067  192.168.0.11  8.8.8.8       DNS          75 Standard query 0xd7c8 AAAA ssl.gstatic.com
100 1.087281  8.8.8.8       192.168.0.11  DNS         103 Standard query response 0xd7c8 AAAA ssl.gs

▸ Frame 5: 90 bytes on wire (720 bits), 90 bytes captured (720 bits)
▸ Ethernet II, Src: Vmware_f0:01:58 (00:50:56:f0:01:58), Dst: PcsCompu_23:b2:31 (08:00:27:23:b2:31)
▸ Internet Protocol Version 4, Src: 8.8.8.8, Dst: 192.168.0.11
▾ User Datagram Protocol, Src Port: 53, Dst Port: 44188
    Source Port: 53
    Destination Port: 44188
    Length: 56
    Checksum: 0x82ca [unverified]
    [Checksum Status: Unverified]
    [Stream index: 1]
▸ Domain Name System (response)
```

b. In the Ethernet II frame for the DNS response, what device is the source MAC address and what device is the destination MAC address?

____

c. Notice the source and destination IP addresses in the IP packet.

What is the destination IP address?

____

What is the source IP address?

____

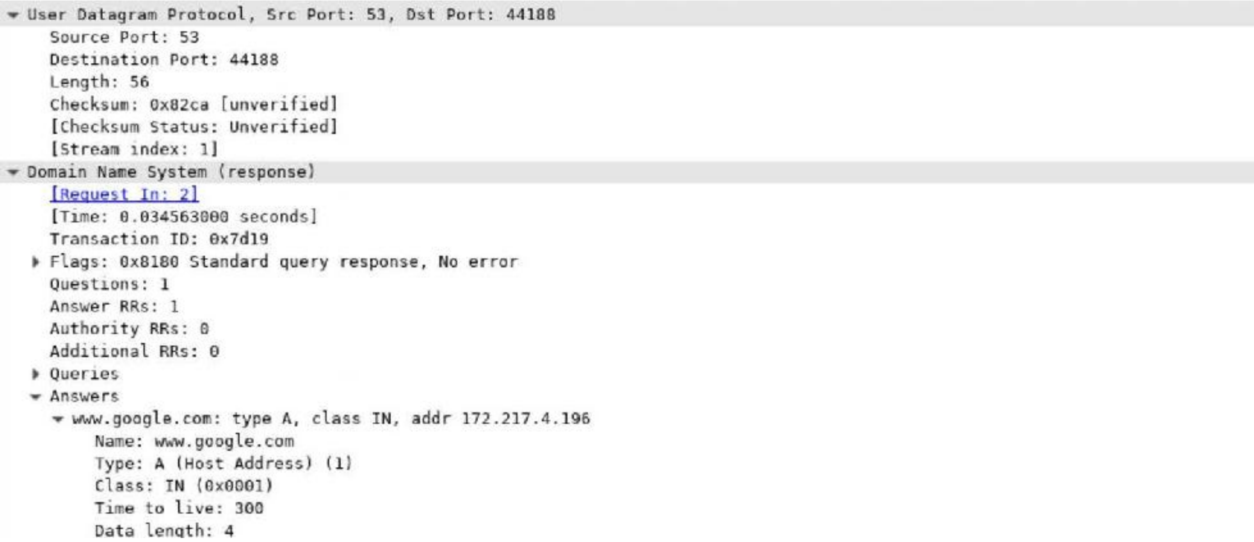
What happened to the roles of source and destination for the VM and default gateway?

____

d. In the UDP segment, the role of the port numbers has also reversed. The destination port number is 44188. Port number 44188 is the same port that was generated by the VM when the DNS query was sent to the DNS server. Your VM listens for a DNS response on this port.

The source port number is 53. The DNS server listens for a DNS query on port 53 and then sends a DNS response with a source port number of 53 back to the originator of the DNS query.

When the DNS response is expanded, notice the resolved IP addresses for www.google.com in the **Answers** section.



```
▾ User Datagram Protocol, Src Port: 53, Dst Port: 44188
    Source Port: 53
    Destination Port: 44188
    Length: 56
    Checksum: 0x82ca [unverified]
    [Checksum Status: Unverified]
    [Stream index: 1]
▾ Domain Name System (response)
    [Request In: 2]
    [Time: 0.034563000 seconds]
    Transaction ID: 0x7d19
  ▸ Flags: 0x8180 Standard query response, No error
    Questions: 1
    Answer RRs: 1
    Authority RRs: 0
    Additional RRs: 0
  ▸ Queries
  ▾ Answers
    ▾ www.google.com: type A, class IN, addr 172.217.4.196
        Name: www.google.com
        Type: A (Host Address) (1)
        Class: IN (0x0001)
        Time to live: 300
        Data length: 4
```

Reflection Question

What are the benefits of using UDP instead of TCP as a transport protocol for DNS?

____

____