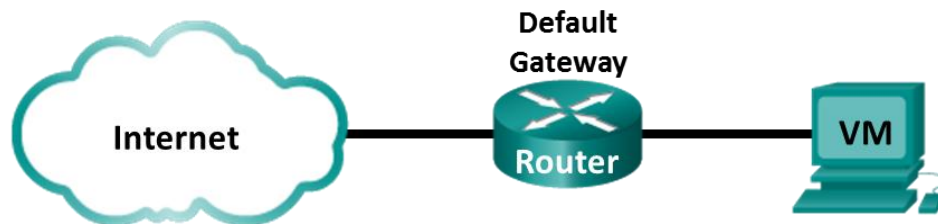


Lab - Using Wireshark to Examine a UDP DNS Capture

Topology



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.

Required Resources

- CyberOps Workstation virtual machine
- Internet access

Instructions

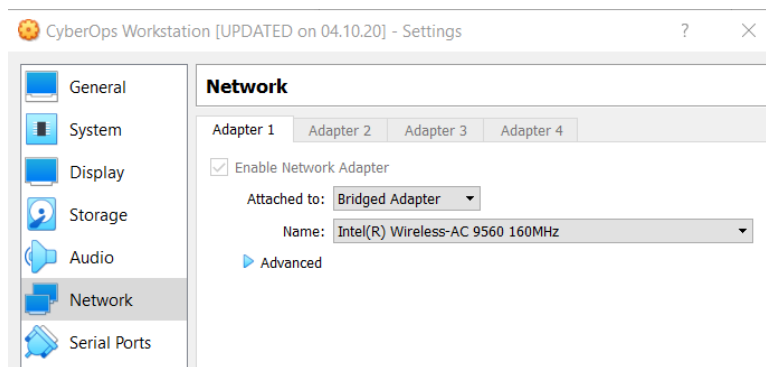
Part 1: Record VM's IP Configuration Information

In Part 1, you will use commands on your CyberOps 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	NAT - 192.168.157.133, Bridged - 192.168.0.24
MAC address	00:0c:29:d3:f6:54
Default gateway IP address	NAT - 192.168.157.2, Bridged - 192.168.0.1

Description	Settings
DNS server IP address	NAT - 8.8.4.4, Bridged - 193.210.19.19

- a. Your CyberOps Workstation VM network settings should be set to bridged adapter. To check your network settings go to: **Machine > Settings**, select **Network**, the tab Adapter 1, Attached to: **Bridged Adapter**.



- b. Open a terminal in the VM. Enter **ifconfig** at the prompt to display interface information. If you do not have an IP address on your local network, run the following command in the terminal:

```
[analyst@secOps ~]$ sudo lab.support.files/scripts/configure_as_dhcp.sh
Configuring the NIC to request IP info via DHCP...
Requesting IP information...
IP Configuration successful.
```

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>
```

- c. 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
```

- d. 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
```

Lab - Using Wireshark to Examine a UDP DNS Capture

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.

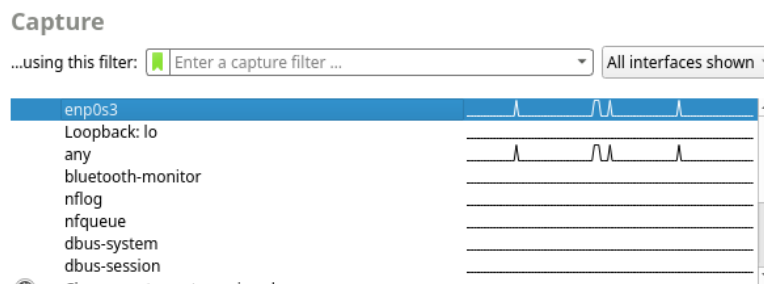
Part 2: Use Wireshark to Capture DNS Queries and Responses

In Part 2, you will set up Wireshark to capture DNS query and response packets. This will demonstrate the use of the UDP transport protocol while communicating with a DNS server.

- In the terminal window, start Wireshark and click **OK** when prompted.

```
[analyst@secOps ~]$ wireshark &
```

- In the Wireshark window, select and double-click **enp0s3** from the interface list.



- Open the web browser and navigate to **www.google.com**.
- Click **Stop** to stop the Wireshark capture when you see Google's home page.

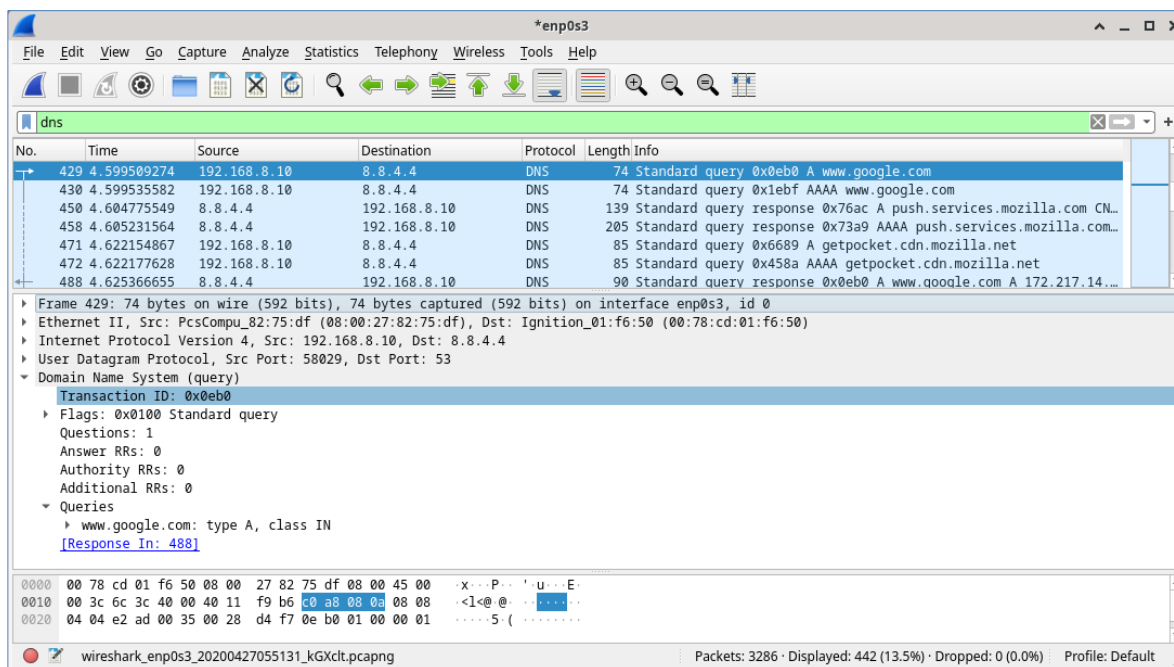
Part 3: Analyze Captured DNS or UDP Packets

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.

- In the Wireshark main window, type **dns** in the **Filter** field. Click **Apply**.

Note: If you do not see any results after the DNS filter was applied, close the web browser. In the terminal window, type ping **www.google.com** as an alternative to the web browser.



- 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 429 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.

- In the first line in the packet details pane, frame 429 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.
- 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?

yes

- In the Internet Protocol Version 4 line, the IP packet Wireshark capture indicates that the source IP address of this DNS query is 192.168.8.10 and the destination IP address is 8.8.4.4. 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?

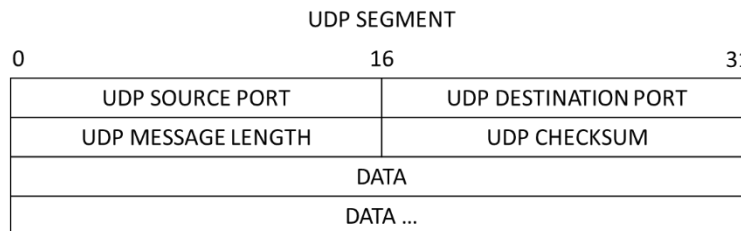
Device	IP Address	MAC Address
Source Workstation	192.168.0.24	00:0c:29:d3:f6:54
Destination DNS Server/ Default Gateway	193.210.19.19	f4:c1:14:85:61:f6

Lab - Using Wireshark to Examine a UDP DNS Capture

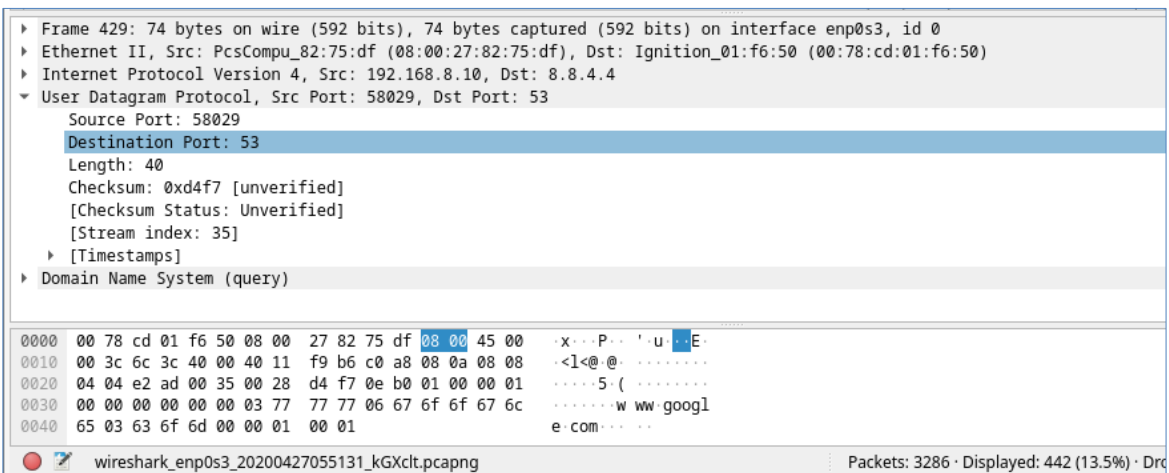
Note: The destination IP address is for the DNS Server, but the destination MAC address is for the default gateway.

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.

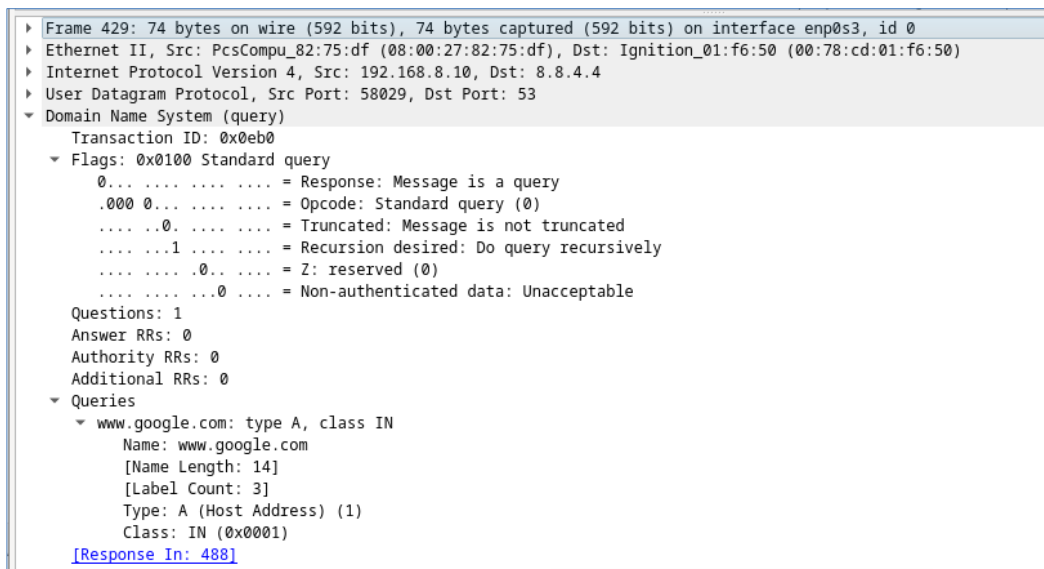


Click the arrow next to User Datagram Protocol to view the details. Notice that there are only four fields. The source port number in this example is 58029. The source port was randomly generated by the VM using port numbers that are not reserved. The destination port is 53. Port 53 is a well-known port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



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.



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	74 bytes
Source MAC address	00:0c:29:d3:f6:54
Destination MAC address	f4:c1:14:85:61:f6
Source IP address	192.168.0.24
Destination IP address	193.210.19.19
Source port	56856
Destination port	53

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

yes after adapter was switched to bridged

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

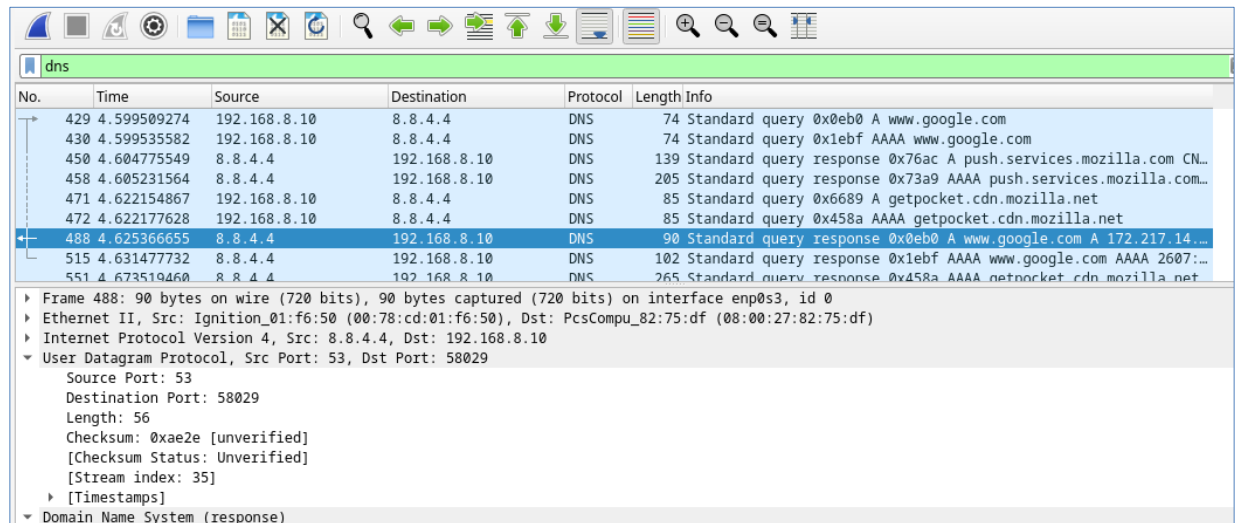
no

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.

Lab - Using Wireshark to Examine a UDP DNS Capture

- a. In this example, frame 488 is the corresponding DNS response packet. Notice the number of bytes on the wire is 90. 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.



No.	Time	Source	Destination	Protocol	Length	Info
429	4.599509274	192.168.8.10	8.8.4.4	DNS	74	Standard query 0x0eb0 A www.google.com
430	4.599535582	192.168.8.10	8.8.4.4	DNS	74	Standard query 0x1ebf AAAA www.google.com
450	4.604775549	8.8.4.4	192.168.8.10	DNS	139	Standard query response 0x76ac A push.services.mozilla.com CN...
458	4.605231564	8.8.4.4	192.168.8.10	DNS	205	Standard query response 0x73a9 AAAA push.services.mozilla.com...
471	4.622154867	192.168.8.10	8.8.4.4	DNS	85	Standard query 0x6689 A getpocket.cdn.mozilla.net
472	4.622177628	192.168.8.10	8.8.4.4	DNS	85	Standard query 0x458a AAAA getpocket.cdn.mozilla.net
488	4.625366655	8.8.4.4	192.168.8.10	DNS	90	Standard query response 0x0eb0 A www.google.com A 172.217.14...
515	4.631477732	8.8.4.4	192.168.8.10	DNS	102	Standard query response 0x1ebf AAAA www.google.com AAAA 2607:...
551	4.673519460	8.8.4.4	192.168.8.10	DNS	265	Standard query response 0x458a AAAA netpocket.cdn.mozilla.net

Frame 488: 90 bytes on wire (720 bits), 90 bytes captured (720 bits) on interface enp0s3, id 0

Ethernet II, Src: Ignition_01:f6:50 (00:78:cd:01:f6:50), Dst: PcsCompu_82:75:df (08:00:27:82:75:df)

Internet Protocol Version 4, Src: 8.8.4.4, Dst: 192.168.8.10

User Datagram Protocol, Src Port: 53, Dst Port: 58029

Source Port: 53
Destination Port: 58029
Length: 56
Checksum: 0xae2e [unverified]
[Checksum Status: Unverified]
[Stream index: 35]
[Timestamps]

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?
source - f4:c1:14:85:61:f6
dest - 00:0c:29:d3:f6:54

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

What is the destination IP address? 193.210.19.19

What is the source IP address? 192.168.0.24

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

- d. In the UDP segment, the role of the port numbers has also reversed. The destination port number is 58029. Port number 58029 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.

Lab - Using Wireshark to Examine a UDP DNS Capture

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

The image shows a Wireshark packet capture of a DNS response. The top section displays a list of packets, with packet 488 selected. This packet is a DNS response from 192.168.8.10 to 8.8.4.4. The bottom section shows the expanded details of this packet, specifically the 'Domain Name System (response)' section. The transaction ID is 0x0eb0. The flags indicate a standard query response with no error. The response contains one question and one answer. The answer is for the domain 'www.google.com', type A, class IN, with an address of 172.217.14.196. The time to live is 37 seconds, and the data length is 4 bytes. A link to the request in packet 429 is provided.

No.	Time	Source	Destination	Protocol	Length	Info
472	4.622177628	192.168.8.10	8.8.4.4	DNS	85	Standard query 0x458a AAAA getpocket.cdn.mozilla.net
488	4.625366655	8.8.4.4	192.168.8.10	DNS	90	Standard query response 0x0eb0 A www.google.com A 172.217.14.196
515	4.631477732	8.8.4.4	192.168.8.10	DNS	102	Standard query response 0x1ebf AAAA www.google.com AAAA 2607:...
551	4.673519460	8.8.4.4	192.168.8.10	DNS	265	Standard query response 0x458a AAAA getpocket.cdn.mozilla.net...

▼ Domain Name System (response)
Transaction ID: 0x0eb0
▼ Flags: 0x8180 Standard query response, No error
1... .. = Response: Message is a response
.000 0... .. = Opcode: Standard query (0)
... .. = Authoritative: Server is not an authority for domain
... .. = Truncated: Message is not truncated
... ..1 ... = Recursion desired: Do query recursively
... ..1 ... = Recursion available: Server can do recursive queries
... ..0... .. = Z: reserved (0)
... ..0... .. = Answer authenticated: Answer/authority portion was not authenticated by the server
... ..0... .. = Non-authenticated data: Unacceptable
... ..0000 = Reply code: No error (0)
Questions: 1
Answer RRs: 1
Authority RRs: 0
Additional RRs: 0
► Queries
▼ Answers
▼ www.google.com: type A, class IN, addr 172.217.14.196
Name: www.google.com
Type: A (Host Address) (1)
Class: IN (0x0001)
Time to live: 37 (37 seconds)
Data length: 4
Address: 172.217.14.196
[\[Request in: 429\]](#)
[Time: 0.025857381 seconds]

Reflection Question

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