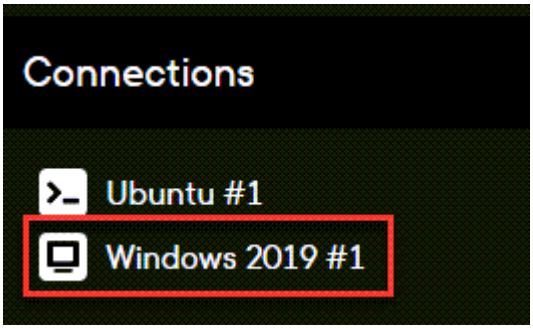
Creating Normal Network Traffic

The purpose of this lab is to introduce you to how the TCP and UDP protocols operate. This lab focuses on the fundamentals of analyzing the protocol behavior for security professionals.

Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) are core functions for how packets function and allowing applications to communicate effectively. You are going to start by generating some standard web traffic to get a better understanding on how these transport protocols function when used normally.

This first challenge will be using the Windows endpoint.



Important Note: We are building a custom environment just for you! This may take a few minutes depending on its complexity.

- On the desktop, there is a shortcut called "STATUS". Double-clicking this will give you an overview on applications installing and/or lab files being downloaded. This does not update automatically - you will need to close and reopen to get an updated progress status. For most applications, an icon will appear on the desktop once the installation is complete!

While you wait, check out some relevant resources:

- Transport Layer
- Transmission Control Protocol (TCP)
- User Datagram Protocol (UDP)

1. You'll want to start by opening Wireshark via the shortcut on the desktop and double clicking the "*Ethernet*" interface.

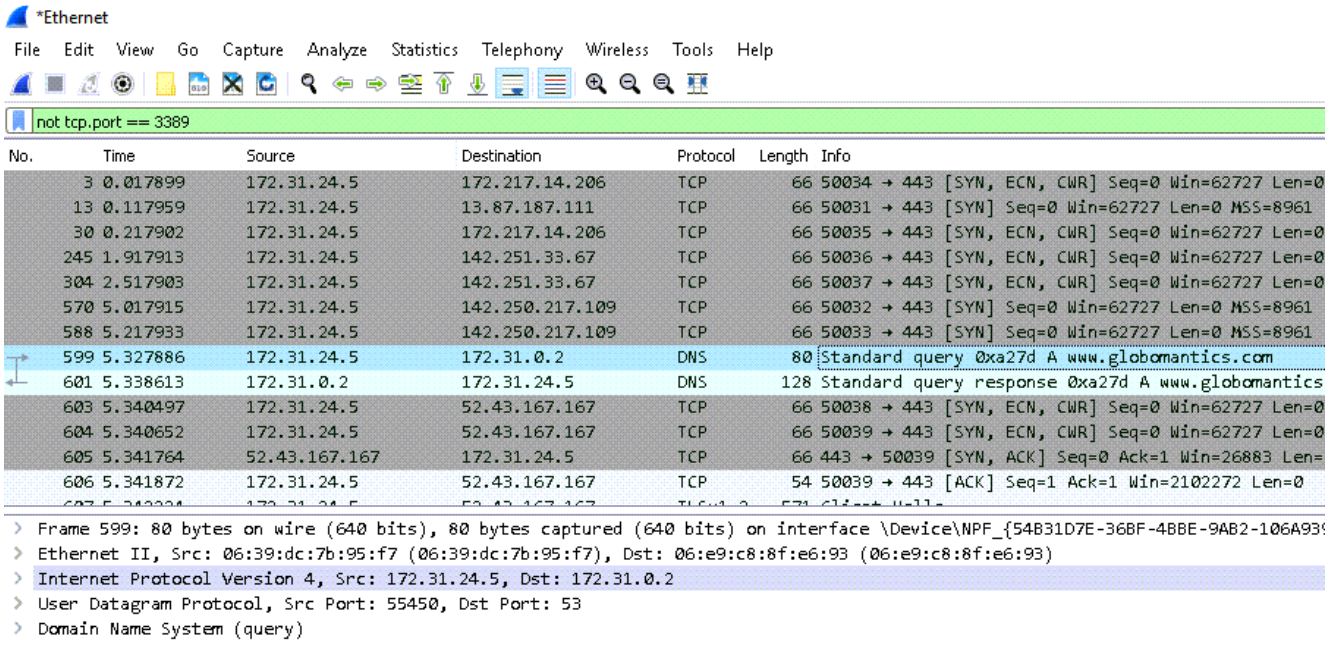
Note: This will allow you to capture live traffic you are about to generate. Leave this up and running in the background.

2. Next, open up Firefox and browse to: `https://www.globomantics.com`. If you encounter any certificate errors, click "*Show Advanced*" and "*Proceed to www.globomantics.com*".

3. Return to Wireshark and stop the capture so you can inspect the traffic.

   Analysis: How many packets do you think you generated with that single request? A simple web connection may generate more packets than initially anticipated.

4. Since you have an active RDP session to the machine, you'll need to filter that traffic out. Add a filter for `not tcp.port == 3389`.



5. What's left over? You'll notice some packets near the top for "DNS". By default, DNS uses UDP for transport, so click one of those packets.

   Note: When you browse to a domain (ex. `www.globomantics.com`), the domain must first be translated into an IP address. The protocol that allows for this translation to happen is DNS, or Domain Name System.

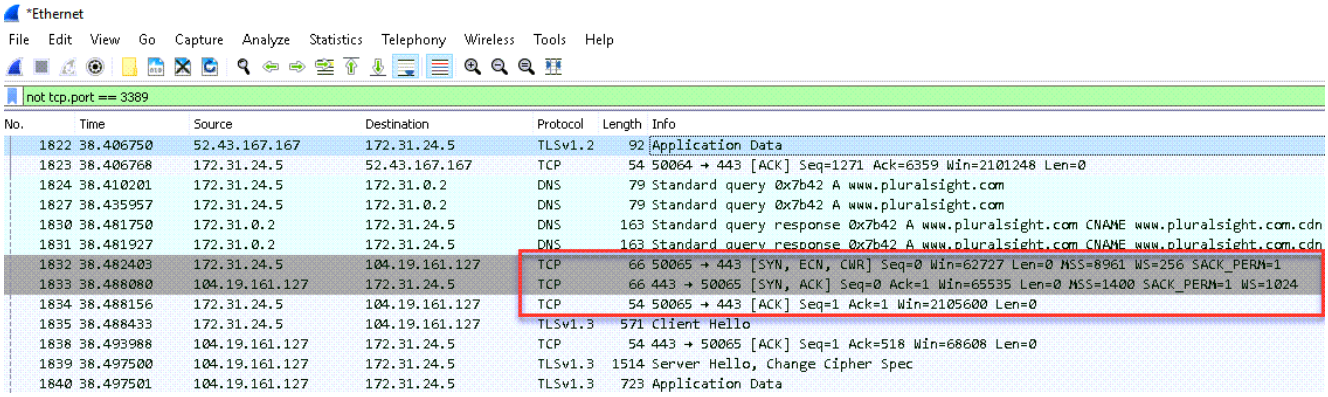
6. Expand the "User Datagram Protocol" (UDP) section in Wireshark to see the individual fields. Here you can see there are only four fields: source port, destination port, length, and checksum. UDP is considered a "connectionless" transport protocol, meaning there are no built-in methods for confirming packets successfully reached their destination. This allows for the overhead on packets to be much smaller which is perfect for applications like DNS.

```
> Frame 599: 80 bytes on wire (640 bits), 80 bytes captured
> Ethernet II, Src: 06:39:dc:7b:95:f7 (06:39:dc:7b:95:f7), D
> Internet Protocol Version 4, Src: 172.31.24.5, Dst: 172.31
v User Datagram Protocol, Src Port: 55450, Dst Port: 53
    Source Port: 55450
    Destination Port: 53
    Length: 46
    Checksum: 0x7085 [unverified]
    [Checksum Status: Unverified]
    [Stream index: 0]
  > [Timestamps]
    UDP payload (38 bytes)
> Domain Name System (query)
```

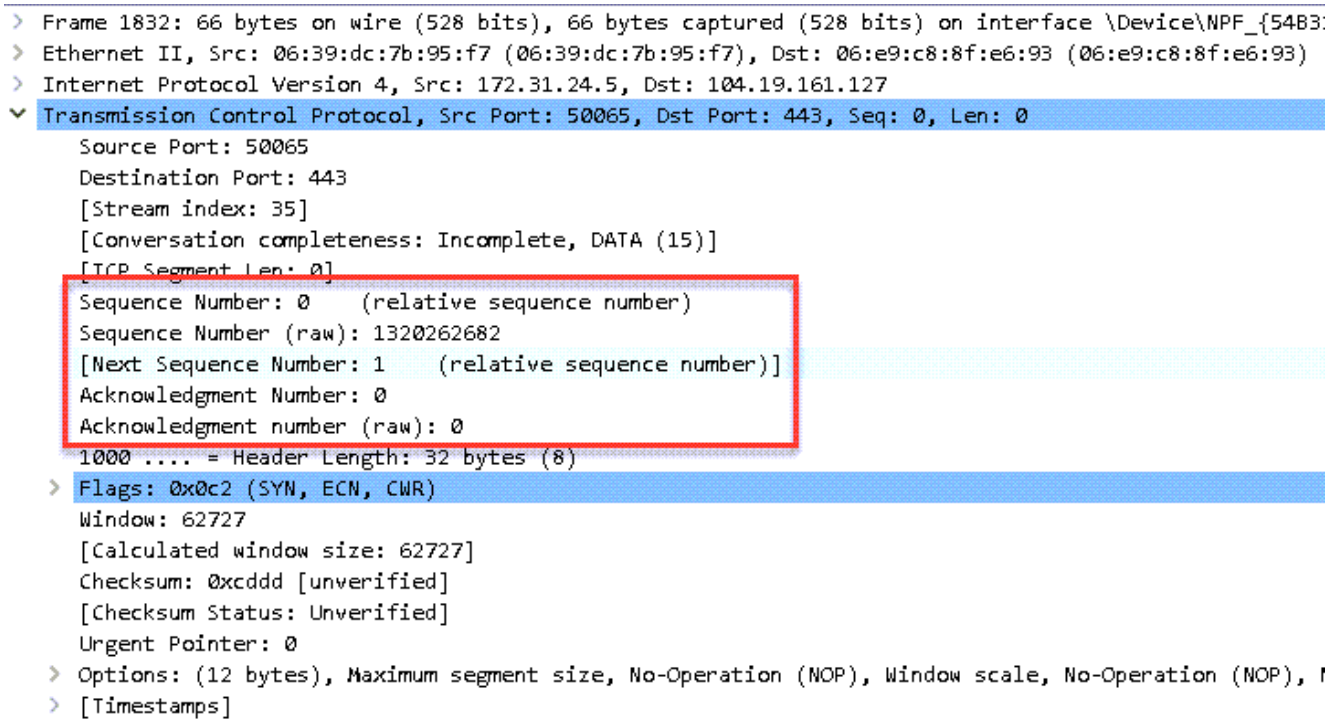
Analysis: It is important to understand what protocols use TCP and UDP by default. This will help you find anomalies throughout your network traffic. For example, DNS can use TCP, but it only does so when sending large payloads. When used legitimately, this might be a DNS Zone transfer. When use maliciously, this is how you might catch DNS tunneling or exfiltration.

7. In contrast, the web connection will be using TCP for transport. Scroll down in Wireshark until you see TCP packets with a destination port of 443. This should be right after the DNS packets.

8. Taking a quick glance at this list will reveal three TCP packets followed by packets marked as "TLSv1.3" mixed with additional TCP packets.



9. Click on one of the TCP packets and expand the details pane to reveal all the fields. Compared to UDP, there are a lot more. Near the top you can see fields for sequence and acknowledgement numbers. TCP is considered a "connection-oriented" transport protocol and it uses these fields to guarantee every packet is received and the data is pieced together in the correct order. This is a necessary overhead to have when ensuring the delivery of a web page.



Now that you understand the fundamental difference between these two transport protocols, you can move on to understanding the how specific fields may be used for malicious purposes or how you might detect anomalies when these protocols don't behave as expected.

It is important to note, while many packets may be marked as "TLSv1.3" or some other application protocol, TCP or UDP will still be used for transport and the data will still exist for all relevant packets.