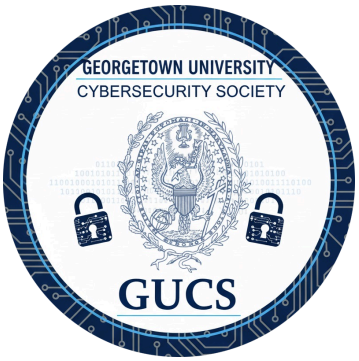
Intro to Packet Structure and IP Addressing

(Short pre-lab primer – read before Lab #1; ~10–15 min)



1. What is a network packet?

A packet is a unit of data sent across a network. When you visit a website, send an email, or stream video, the data is broken into packets. Each packet contains:

- **Header(s)** — metadata used by network devices (addresses, protocol flags, length, checksums, etc.)
- **Payload** — the actual data (e.g., part of an HTTP response, DNS answer)
- **Trailer** (sometimes) — error-checking info (e.g., Ethernet FCS)

Common encapsulation (outer → inner):

Ethernet frame → IP packet (IPv4 or IPv6) → TCP/UDP segment → Application data

⚠ IPv6 adds some differences

Instead of IPv4 headers with checksums, IPv6 uses:

- Fixed 40-byte header
- No header checksum
- Optional extension headers (routing, fragmentation, security, etc.)

2. Key header fields you'll see in Wireshark

- **Ethernet**: source MAC, destination MAC, EtherType
- **IP (IPv4)**: source IP, destination IP, TTL, protocol (TCP/UDP/ICMP), header checksum
- **TCP**: source port, dest port, sequence number, acknowledgment number, flags (SYN, ACK, FIN, RST), window size
- **UDP**: source/dest port, length, checksum
- **HTTP (application)**: method (GET/POST), path, headers, status codes

Tip: in Wireshark you can expand each layer to inspect these fields and use **"Follow TCP Stream"** to reassemble application-level text.

3. Very short IP addressing basics

- **IPv4** format: a.b.c.d (e.g., 192.168.1.10) — 32 bits

- **Network vs host**: an address is split by a subnet mask (e.g., /24 means first 24 bits are the network)
- **Private vs public**: private ranges (e.g., 10.0.0.0/8, 192.168.0.0/16, 172.16.0.0/12) are not routable on the public internet
- **What to watch for in PCAPs**: look for source/destination IP pairs, unusual ports, or malformed packets

4. Quick look at TCP three-way handshake

When establishing a TCP connection:

1. **SYN** — client asks to start a connection (SYN flag set)
2. **SYN-ACK** — server acknowledges and agrees (SYN + ACK set)
3. **ACK** — client acknowledges the acknowledgement → connection established
   In Wireshark, filter tcp.flags.syn==1 && tcp.flags.ack==0 to find initial SYNs.

5. Practical tips for the lab

- Use filters: ip.addr == x.x.x.x, tcp.port == 80, http, dns
- Right-click packet → **Follow** → **TCP Stream** to read HTTP text in sequence
- Export objects: **File** → **Export Objects** → **HTTP** to pull files transferred in the capture

6. Challenge hint (no spoilers)

If the challenge flag is inside HTTP traffic, search for http.request and look at the request URIs or headers. If it's in DNS, search dns and inspect TXT records or query names.

7. TryHackMe

https://tryhackme.com/room/networksecurityprotocols
https://tryhackme.com/room/wiresharkthebasics
https://tryhackme.com/room/wiresharkpacketoperations
https://tryhackme.com/room/wiresharktrafficanalysis