CYBERSECURITY DAILY DIARY

DAY-20
DATE- 12 July, 2025

Topics Covered:

- Wireshark deep dive: use display filters, follow TCP streams
- Analyze DHCP/HTTP flows.

What I Did Today

Display Filters These are essential for cutting through packet noise. Unlike capture filters (which limit what gets recorded), display filters let you sift through captured data with surgical precision.

- Examples I used:
  - `http` — shows only HTTP traffic
  - `bootp` — reveals DHCP packets (BOOTP is the legacy name)
  - `tcp.port == 80` — isolates HTTP over TCP
  - `ip.addr == 192.168.56.101` — focuses on a specific VM
- Learned that filters are case-sensitive and must be syntactically correct — Wireshark highlights errors in red.

Follow TCP Stream Right-clicking a TCP packet and selecting "Follow TCP Stream" reconstructs the entire conversation between client and server.

- Used it to view a full HTTP GET request and the corresponding HTML response — seeing raw headers and payloads made protocol behavior much clearer.
- Realized how useful this is for debugging login forms, API calls, or even spotting credentials in plaintext (if not encrypted).

Analyzing DHCP Flows Observed the classic **DORA** process:

- **D**iscover
- **O**ffer
- **R**equest
- **A**cknowledge
- Each step is visible in the packet list with `bootp` filter.
- Noticed how the client uses 0.0.0.0 as its source IP initially and how the server responds with an IP lease offer.
- Reinforced how DHCP relies on broadcast and UDP (ports 67/68).

Analyzing HTTP Flows

- Tracked a browser request to a local web server.
- Saw the TCP handshake (`SYN`, `SYN-ACK`, `ACK`), followed by the HTTP GET request.
- Observed status codes like `200 OK` and headers like `Content-Type`, `User-Agent`, and `Set-Cookie`.
- Realized how much metadata is exchanged before the actual content — and how vulnerable it can be without HTTPS.