**🧩 1. OSI Model & TCP/IP Stack**

**🔹 OSI Model (7 Layers)**

The OSI (Open Systems Interconnection) model is a conceptual framework that standardizes how data moves across a network.

| **Layer** | **Function** | **Example** |
| --- | --- | --- |
| 7. Application | Interfaces for user apps | HTTP, FTP, DNS |
| 6. Presentation | Data formatting/encryption | SSL/TLS |
| 5. Session | Manages sessions | TCP session control |
| 4. Transport | Reliable delivery | TCP/UDP |
| 3. Network | Routing & addressing | IP, ICMP |
| 2. Data Link | Frame transfer | MAC, ARP |
| 1. Physical | Hardware transmission | Ethernet, Wi-Fi |

🔧 **Use Case**: If a cyberbullying message is sent via Instagram, it travels through all these layers—from app interface to physical transmission.

**🔹 TCP/IP Stack**

A more practical 4-layer model used in real-world networking:

* **Application**: HTTP, DNS
* **Transport**: TCP, UDP
* **Internet**: IP, ICMP
* **Network Access**: Ethernet, Wi-Fi

**🌐 2. Common Protocols**

**🔸 HTTP/HTTPS**

* Used for web communication
* HTTPS adds encryption via SSL/TLS
* Vulnerable to phishing, fake websites, and man-in-the-middle attacks

**🔸 DNS**

* Resolves domain names to IP addresses
* Can be abused for **DNS spoofing** or **DNS tunneling** (data exfiltration)

**🔸 TCP**

* Reliable, connection-oriented
* Used in messaging apps, email, file transfers
* Vulnerable to **SYN flood attacks**

**🔸 UDP**

* Fast, connectionless
* Used in gaming, video streaming
* Vulnerable to **DDoS amplification**

**📄 3. IPDR Logs: Structure, Interpretation, and Anomaly Detection**

**🔹 What Are IPDR Logs?**

**Internet Protocol Detail Records**—logs generated by ISPs or telecoms that record:

* Source & destination IPs
* Ports used
* Timestamps
* Protocol type
* Duration of session

**🔹 How to Interpret:**

* Identify frequent connections to suspicious IPs
* Flag unusual port usage (e.g., port 4444 for remote shells)
* Detect spikes in traffic or odd time-of-day activity

**🔹 Anomaly Detection:**

Use Python + Pandas + Regex to:

* Parse logs
* Group by user/IP
* Detect patterns like repeated access to known phishing domains

**🛡️ 4. Cybercrime Categories**

**🔸 Phishing**

* Fake websites or messages tricking users into revealing credentials
* Often uses spoofed domains and urgent language

**🔸 Doxxing**

* Publishing private info (address, phone, etc.) without consent
* Common in online harassment cases

**🔸 Cyberbullying**

* Repeated online abuse, threats, or humiliation
* Can lead to mental health issues or even suicide

🔍 **Your Tool’s Role**: Detect keywords, patterns, and sentiment in scraped content that match these categories.

**🔬 5. Network Forensics Basics**

**🔹 What It Is:**

The process of capturing, analyzing, and reconstructing network traffic to investigate incidents.

**🔹 Key Techniques:**

* **Packet capture**: Using tools like Wireshark or tcpdump
* **Log correlation**: Matching IPDR logs with app behavior
* **Timeline reconstruction**: What happened, when, and how

**🔹 Use Case:**

If a suicide is linked to an online game, forensic analysis might reveal:

* Chat logs
* Game server traffic
* DNS queries to game-related domains

Want to build a mini IPDR parser or simulate a phishing detection module using NLP? I can help you scaffold that next.

Great follow-up, Anchit. The **TCP/IP stack** is the practical backbone of how data moves across the internet—used in real-world systems, unlike the more conceptual OSI model. Let’s break it down layer by layer and show how it connects to cybercrime detection, IPDR logs, and your project’s goals.

**🌐 TCP/IP Stack: The 4-Layer Real-World Networking Model**

The TCP/IP model simplifies the OSI model into **four layers**, each with specific responsibilities. It’s what your devices actually use to communicate over the internet.

| **Layer** | **Role** | **Protocols** | **Cybercrime Relevance** |
| --- | --- | --- | --- |
| 4️⃣ Application | Interfaces for user apps | HTTP, HTTPS, DNS, FTP, SMTP | Phishing, cyberbullying, fake websites |
| 3️⃣ Transport | Reliable data delivery | TCP, UDP | DDoS, SYN floods, port scanning |
| 2️⃣ Internet | Routing & addressing | IP, ICMP | IP spoofing, geolocation, IPDR logs |
| 1️⃣ Network Access | Physical & link-level transmission | Ethernet, Wi-Fi, ARP | MAC spoofing, packet sniffing |

**🔍 Layer-by-Layer Breakdown**

**🔹 1. Network Access Layer**

* Handles physical transmission (cables, Wi-Fi) and MAC addressing.
* Cybercrime angle: **MAC spoofing**, **packet sniffing**, **unauthorized access** via Wi-Fi.

**🔹 2. Internet Layer**

* Responsible for routing packets across networks using IP addresses.
* Key protocols: **IP**, **ICMP** (used in ping/traceroute).
* Cybercrime angle: **IP spoofing**, **IPDR log analysis**, **geolocation tracking**.

**🔹 3. Transport Layer**

* Ensures reliable or fast delivery of data.
* **TCP**: Reliable, ordered delivery (used in messaging, email).
* **UDP**: Fast, connectionless (used in gaming, streaming).
* Cybercrime angle: **SYN flood attacks**, **port scanning**, **malware payload delivery**.

**🔹 4. Application Layer**

* Where user-facing apps live: browsers, email clients, social media.
* Protocols: **HTTP/HTTPS**, **DNS**, **FTP**, **SMTP**.
* Cybercrime angle: **Phishing**, **fake websites**, **malicious downloads**, **cyberbullying**.

**🧠 Why It Matters for Your Project**

Your tool will likely operate at the **Application Layer**, scraping and analyzing content from websites, forums, and social media. But understanding the **Transport and Internet layers** helps you:

* Interpret **IPDR logs** (source/destination IP, protocol used)
* Detect **network anomalies** (e.g., repeated access to shady domains)
* Map **crime patterns** to specific protocols (e.g., UDP-based gaming linked to suicide cases)

**🛠️ Example: Mapping a Cybercrime Incident**

Let’s say a suicide is linked to an online game:

* **Application Layer**: Game chat logs show bullying
* **Transport Layer**: UDP used for fast game traffic
* **Internet Layer**: IPDR logs show frequent access to game servers
* **Network Access Layer**: MAC address identifies the victim’s device

Want to simulate a TCP handshake, parse IPDR logs, or build a protocol classifier in Python? I can help you scaffold that next.