AZERBAIJAN CYBERSECURITY CENTER

FINAL PROJECT

Log Analysis & CTI

TECHNION
Azrieli School of Continuing Studies of the Technion

OBJECTIVES

- Build a Python tool that reads **web server logs** and **automatically** finds possible **security threats**.

- Practice key skills: **file parsing**, **linking data**, and using outside **Cyber Threat Intelligence (CTI)** to assess risk.

- Use **AI** to turn technical log data into clear, short reports that help a SOC team in Azerbaijan act quickly.

STAGE 1 – PARSE THE LOG

1. Run the script from the command line with the path to **`access.log`**. If the file is missing, print a friendly error and exit safely.

2. Read each line and get: **Source IP**, **Timestamp**, **HTTP Method**, **Status Code**. If a line is broken, skip it and keep going.

3. Count per-IP activity: total requests and how many are **4xx** errors. Keep a list of unique IPs.

STAGE 2 – CHECK IP REPUTATION (CTI)

1. For every unique IP, query one CTI source: **AbuseIPDB** (scrape score, total reports, country) or **Cisco Talos** (scrape web reputation, owner) or **VirusTotal API** (analysis stats).

2. Mark IPs as **suspicious** if they have high risk/poor reputation. Save the CTI details with the source name. Handle network errors gently and continue.

3. **Bonus:** also look at **User-Agent** strings (e.g., **`sqlmap`**, **`Nmap`**, **`Hydra`**). If a suspicious IP uses a malicious agent, flag it as **high priority**.

STAGE 3 – EXPLAIN & REPORT

1. Pick one high-risk IP. Ask an AI to write **one simple sentence** that explains the threat to a non-technical person.

2. Create a **report** for all suspicious IPs: IP, CTI findings, total requests, and 4xx count. Save as .txt (make a **`reports/`** folder if missing).

3. **Bonus:** also save as **Markdown/HTML**, and send whole-log stats (total requests, 404/200 ratio, unique IPs) to AI to describe any **anomalies**.