Student ML Project: Network Traffic Capture, Baseline, and Detection — Full Guide

Compiled conversation with prompt summaries, explanations, proposed lab topology, and references

Prompt summary: Define sources that explain how to baseline 'normal' network traffic for detecting anomalies/malicious activity (not coding-focused).

Baseline 'Normal' Traffic for Anomaly/Malicious Detection — Method and Sources

Approach: combine protocol standards (RFCs), real network flow/log evidence, and labeled academic datasets.

RFC anchors for 'normal': IPv4 (RFC 791), UDP (RFC 768), ICMP (RFC 792), TCP (RFC 793/STD 7), DNS (RFC 1035), TLS 1.3 (RFC 8446), QUIC (RFC 9000) and HTTP/3 (RFC 9114), IANA ports (RFC 6335).

Measure 'normal' in practice with flows (IPFIX, RFC 7011) and protocol logs (Zeek/Bro). Build a baseline per protocol/role/time-of-day and flag deviations.

Datasets for validation: CIC-IDS2017, UNSW-NB15, CTU-13, MAWI/MAWILab. Note biases and validate carefully.

Suggested workflow: Map expected services; collect flows/Zeek logs for 2–4 weeks; extract features; detect deviations; validate with labeled datasets.

References & Links:

- RFC 791 — Internet Protocol (IPv4) — https://www.rfc-editor.org/rfc/rfc791
- RFC 768 — User Datagram Protocol (UDP) — https://www.rfc-editor.org/rfc/rfc768
- RFC 792 — Internet Control Message Protocol (ICMP) — https://www.rfc-editor.org/rfc/rfc792
- RFC 793 (STD 7) — Transmission Control Protocol (TCP) — https://www.rfc-editor.org/rfc/rfc793
- RFC 1035 — DNS — https://www.rfc-editor.org/rfc/rfc1035
- RFC 8446 — TLS 1.3 — https://www.rfc-editor.org/rfc/rfc8446
- RFC 9000 — QUIC — https://www.rfc-editor.org/rfc/rfc9000
- RFC 9114 — HTTP/3 — https://www.rfc-editor.org/rfc/rfc9114
- RFC 6335 — IANA Ports — https://www.rfc-editor.org/rfc/rfc6335
- RFC 7011 — IPFIX — https://www.rfc-editor.org/rfc/rfc7011
- Zeek Documentation — https://docs.zeek.org
- CIC-IDS2017 Dataset — https://www.unb.ca/cic/datasets/ids-2017.html
- UNSW-NB15 Dataset — https://research.unsw.edu.au/projects/unsw-nb15-dataset
- CTU-13 Dataset — https://www.stratosphereips.org/datasets-ctu13
- MAWI / MAWILab — https://mawi.wide.ad.jp/mawilab/

—

Prompt summary: What is DPI (Deep Packet Inspection)?

DPI — Concept, Uses, and Modern Constraints

Definition: inspecting not only packet headers but also payload to classify applications, enforce policy, and detect threats.

Modern limitation: with TLS 1.3 and QUIC/HTTP/3, payload and much of the handshake are encrypted, pushing defenders toward metadata/flow analysis.

Privacy/ethics: IETF considers pervasive monitoring an attack; protocol designs aim to protect privacy.

References & Links:

- NIST CSRC Glossary — Deep Packet Inspection — https://csrc.nist.gov/glossary/term/deep_packet_inspection
- RFC 8446 — TLS 1.3 — https://www.rfc-editor.org/rfc/rfc8446
- RFC 9000 — QUIC — https://www.rfc-editor.org/rfc/rfc9000
- RFC 9114 — HTTP/3 — https://www.rfc-editor.org/rfc/rfc9114
- RFC 7258 — Pervasive Monitoring Is an Attack — https://www.rfc-editor.org/rfc/rfc7258
- RFC 8404 — Effects of Pervasive Encryption on Operators — https://www.rfc-editor.org/rfc/rfc8404

—

Prompt summary: Design a simple lab so a student can sniff traffic (including between other hosts). Can they 'just use PCAP'? How to capture beyond local/broadcast?

Lab Topology for Sniffing: SPAN, TAP, and Gateway Capture

Yes, use libpcap/Npcap and tools (tcpdump/Wireshark/Zeek) to capture pcap/pcapng.

To see traffic between other hosts on a switch, you need visibility: SPAN/Port Mirroring, Inline TAP, or capture at the gateway/bridge.

Recommendation: SPAN is quickest; TAP offers highest fidelity; gateway capture is a simple alternative for tiny labs.

References & Links:

- Wireshark File Formats (pcapng) — https://www.wireshark.org/docs/wsug_html_chunked/AppFiles.html
- libpcap / tcpdump — https://www.tcpdump.org/
- Npcap for Windows — https://nmap.org/npcap/
- Cisco SPAN/ERSPAN Overview — https://www.cisco.com/c/en/us/support/docs/switches/catalyst-6500-series-switches/10570-41.html
- Network TAP concepts (Gigamon overview) — https://www.gigamon.com/resources/learn/what-is-a-network-tap.html
- Zeek Deployment Guide — https://docs.zeek.org/en/current/install/
- IPFIX (RFC 7011) — https://www.rfc-editor.org/rfc/rfc7011

—

Prompt summary: Do they use TAP in real life scenarios?

Real-World Use of TAPs vs SPAN

Yes, TAPs are standard in production for security monitoring (IDS/NDR), performance analysis, financial trading networks, telecom backbones, and datacenters.

SPAN is convenient but best-effort; can drop/alter traffic under load. TAPs provide higher fidelity and cleaner separation for monitoring.

Cloud parallel: AWS VPC Traffic Mirroring is a 'virtual tap' for EC2 instances.

References & Links:

- Zeek docs — preferring TAPs when available — https://docs.zeek.org/en/current/install/
- Cisco Nexus and tap aggregation patterns — https://www.cisco.com/
- NIST SP 800-94 (IDPS Guide) — https://csrc.nist.gov/publications/detail/sp/800-94/rev-1/final
- Gigamon — Network TAP best practices — https://www.gigamon.com/
- Garland Technology — SPAN vs TAP — https://www.garlandtechnology.com/

- AWS VPC Traffic Mirroring — https://docs.aws.amazon.com/vpc/latest/mirroring/what-is-traffic-mirroring.html

—

Prompt summary: For a software engineering student: should they implement a TAP? What should the project include?

Project Scope for a Software Engineering Student

Do not implement a TAP (it is hardware). Instead, write a passive sensor that listens on an interface fed by a TAP or SPAN and performs feature extraction + ML.

Ingest options: (A) libpcap collector that stores pcapng or streams features; (B) run Zeek and consume its logs (conn/dns/tls/http).

Feature ideas (payload-agnostic): flow stats, TCP handshake/flags anomalies per RFC 793, DNS behavior per RFC 1035, TLS/QUIC surface (versions, ALPN).

Modeling: start unsupervised (baseline normal, flag outliers); optionally validate with labeled datasets (CIC-IDS2017, UNSW-NB15, CTU-13, MAWI).

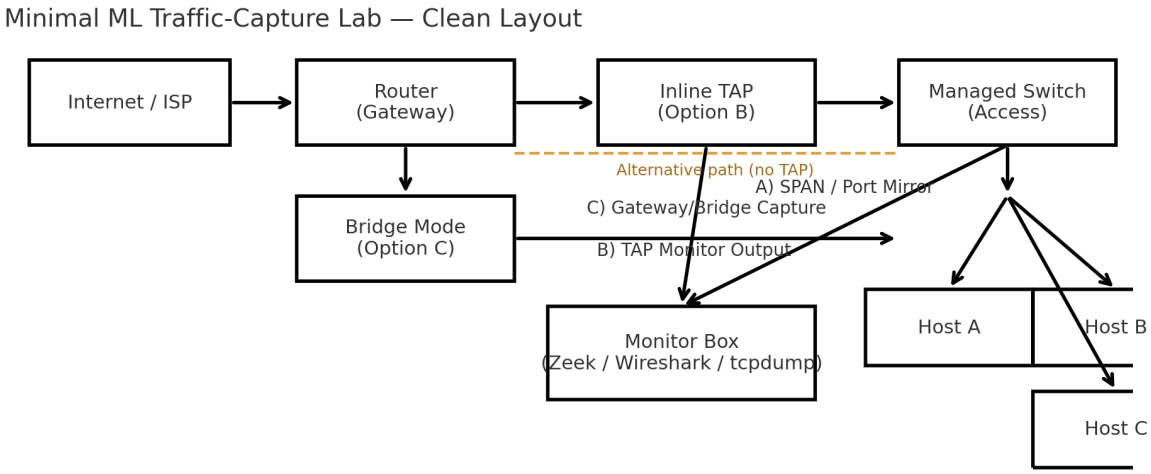
Deliverables: capture plan diagram; collector/Zeek config; feature pipeline; model & evaluation; README with references.

References & Links:

- RFC 793 — TCP — https://www.rfc-editor.org/rfc/rfc793
- RFC 1035 — DNS — https://www.rfc-editor.org/rfc/rfc1035
- RFC 8446 — TLS 1.3 — https://www.rfc-editor.org/rfc/rfc8446
- RFC 9000 — QUIC — https://www.rfc-editor.org/rfc/rfc9000
- RFC 7011 — IPFIX — https://www.rfc-editor.org/rfc/rfc7011
- Zeek Logs Reference — https://docs.zeek.org/en/current/script-reference/log-files.html
- CIC-IDS2017 — https://www.unb.ca/cic/datasets/ids-2017.html
- UNSW-NB15 — https://research.unsw.edu.au/projects/unsw-nb15-dataset
- CTU-13 — https://www.stratosphereips.org/datasets-ctu13
- MAWI/MAWILab — https://mawi.wide.ad.jp/mawilab/

—

Figure 1 — Minimal ML Traffic-Capture Lab



Legend:
• Production path
• Alternative (no TAP)
• Capture points:
  A) SPAN / Port Mirror on switch
  B) Inline TAP monitor output
  C) Gateway/Bridge inline capture

Tips:
• Use A for the quickest setup; use B for highest fidelity.
• Bridge Mode (C) is a simple alternative when no TAP is available.

A: SPAN/Port Mirror, B: Inline TAP monitor output, C: Gateway/Bridge inline capture.