

CYBERSECURITY LECTURE SERIES

Network Security

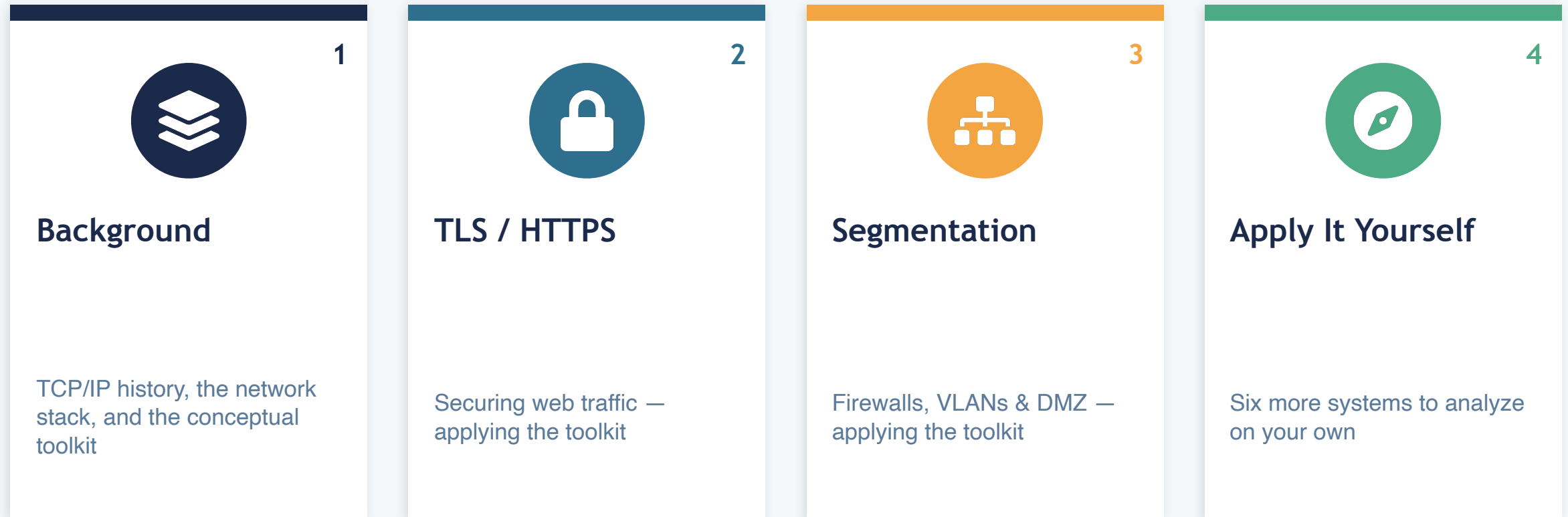
Protecting Traffic, Controlling Flow

A toolkit for analyzing security mechanisms — applied to TLS/HTTPS and network segmentation.

Followup session: guided practice with the toolkit.



From Background to Two Case Studies



Followup session (a few days later): guided practice — trace a TLS handshake, design a DMZ.

An Experiment That Escaped the Lab



The Experiment

Built for ARPANET — a small, closed network of trusted researchers. Goal: create an internetwork connecting heterogeneous networks under a common protocol.



Too Successful

Simple, flexible, ran over almost any physical medium. Adopted everywhere before a replacement could be built. It escaped the lab and became the global standard.



The Consequence

Security had to be layered on top of a global installed base. A trusted research network needed no security; the whole internet inherited that assumption.

The address space parallel: 32-bit IPv4 addresses were chosen because they were “enough for an experiment—even the Defense Department doesn’t need 4.3 billion of anything.” Now wholly inadequate; patched with NAT, being replaced by IPv6. Same pattern, same cause.

How Protocols Are Standardized: IETF and RFCs

The IETF

The Internet Engineering Task Force is the open standards body for internet protocols. First meeting 1986; open to anyone — no membership, no fees. Decisions are made in working groups, not by committee vote.

“Rough consensus and running code”

RFCs — Request for Comments

The name is historical — RFCs are authoritative specifications, not drafts. Freely available at rfc-editor.org; never revised in place: when a protocol is updated, a new RFC obsoletes the old one. Protocol history is fully traceable through RFC lineage.

RFC 1 was published in 1969.

Protocol	RFC #	Year	Notes
IPv4	RFC 791	1981	Original spec; still in use
IPv6	RFC 8200	2017	Obsoletes RFC 2460 (1998)
TCP	RFC 9293	2022	Consolidates RFC 793 + 40 yrs of updates
UDP	RFC 768	1980	3 pages — elegant simplicity
HTTP	RFC 9110	2022	HTTP semantics; transport-independent
TLS 1.3	RFC 8446	2018	Current standard; 1.2 = RFC 5246

rfc-editor.org — free, authoritative, complete back to RFC 1 (1969)

Layering: A Network of Abstractions

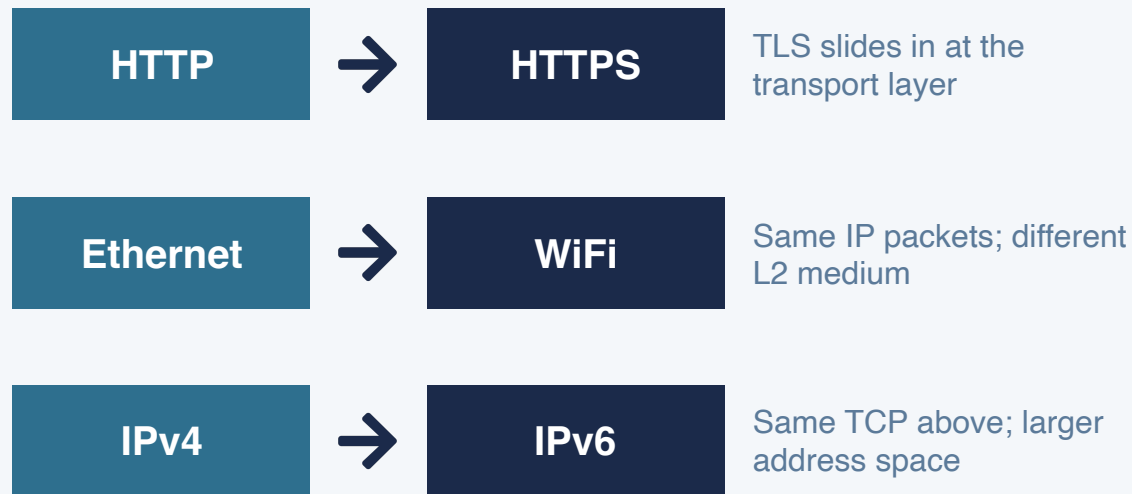
Application	Protocol-specific messaging (HTTP, DNS, SMTP). No layer-level addressing. <i>Security: PGP, DNSSEC — protect message content and lookups</i>
Transport (L4)	End-to-end process communication — TCP (reliable, ordered) or UDP (lightweight). Addresses by port. <i>Security: TLS / SSL — encrypts the connection between client and server</i>
Network (L3)	Routes between heterogeneous networks (IP) — same packet crosses Ethernet, WiFi, fiber. Addresses by IP. <i>Security: IPsec, VPNs, firewalls, subnet isolation</i>
Link (L2)	Local delivery between adjacent nodes (Ethernet, WiFi). Addresses machines by MAC address. <i>Security: MACsec, VLANs — protect traffic on the local segment</i>

Each layer uses the services of the layer below — and can be replaced independently without changing the rest of the stack.

Substitution and Encapsulation

Substitution

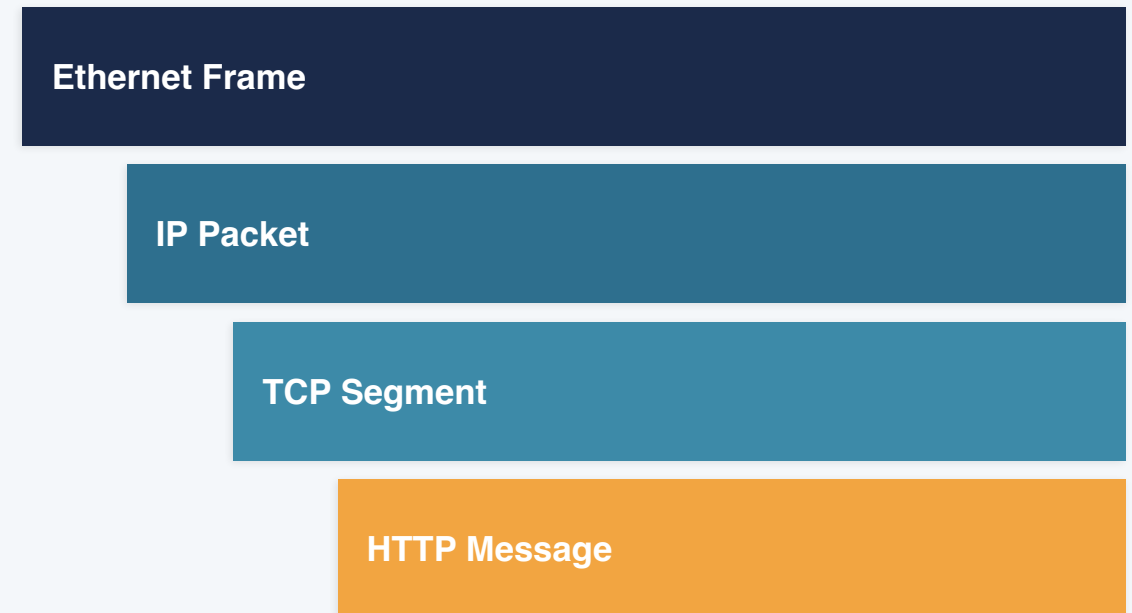
Replace one layer's protocol without changing any other layer.



This is why TLS and IPsec could be added without redesigning the entire stack.

Encapsulation

Each layer wraps the layer above in a header + payload — a Protocol Data Unit (PDU).



Encryption at a layer hides the payload — but outer headers (IP, Ethernet) remain visible.

What Are We Protecting?



Confidentiality

Only intended parties can read the data.

Violation: eavesdropping



Integrity

Data isn't modified in transit without detection.

Violation: tampering



Availability

The system keeps working when legitimate users need it.

Violation: denial of service



Authentication

You actually know who you're talking to.

Violation: impersonation

THE TOOLKIT

Three Questions for Any Security Mechanism

Apply these to TLS, segmentation — or anything you encounter later.

1

What threat does it address?

Which adversary goal — eavesdropping, tampering, impersonation — is this mechanism designed to stop?

2

What's the trust model?

What does the mechanism assume about the attacker, and who/what are we trusting to make it work?

3

What's the residual risk?

Even when it works as intended, what's left exposed? What doesn't this solve?

Defense in Depth



Each layer addresses a different threat:



Perimeter — firewall



Network — segmentation (VLAN/subnet)



Host — endpoint controls



Data — encryption (TLS)

No single layer is sufficient — each addresses a different threat, trust model, and residual risk.

CASE STUDY 1

TLS / HTTPS

Securing the Web

Applying the cryptography toolkit to protect traffic between client and server.



How TLS Protects a Connection

The Handshake (simplified)

- 1 Client & server agree on a cipher suite
- 2 Server presents its certificate (identity + public key)
- 3 Both sides derive a shared session key via Diffie-Hellman
- 4 All further traffic is encrypted and authenticated

Key Concepts

Certificate Authorities

Trusted third parties that vouch for server identity

Chain of Trust

Your browser trusts a CA; the CA vouches for the site

Forward Secrecy

Ephemeral keys mean past sessions can't be decrypted later

TLS Through the Toolkit



Q1

Threat addressed

Eavesdropping, tampering, and impersonation on the path between client and server.



Q2

Trust model

You trust certificate authorities to correctly verify identity before issuing certificates.



Q3

Residual risk

Endpoint compromise, traffic metadata (who's talking to whom), and CA trust failures remain.

CASE STUDY 2

Network Segmentation

Firewalls, VLANs & the DMZ – as One System

Applying the toolkit to controlling where traffic is allowed to flow.



Controlling Where Traffic Can Go



Public-facing servers sit in the DMZ; the firewall on each side enforces different rules for what's allowed to talk to what.



Firewalls

Filter traffic at boundaries by rules (allow/deny, ports, protocols, state).



VLANs

Logically separate traffic on shared hardware — limit broadcast domains.



Subnets / DMZ

Place systems according to trust level; isolate public-facing servers.

Segmentation Through the Toolkit



Q1

Threat addressed

Lateral movement after a breach, and unauthorized access between zones of differing trust.



Q2

Trust model

Zones of trust — public-facing, internal, and data tiers — each assumed to need different protection.



Q3

Residual risk

Encrypted traffic is a blind spot; misconfiguration and insider threats remain — segmentation doesn't help once inside a zone.

APPLY IT YOURSELF

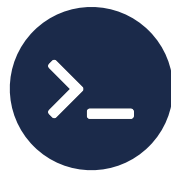
Six More Systems — You Have the Tools

For each, ask: what threat does it address? what's the trust model? what's the residual risk?



PGP

End-to-end email security



SSH

Secure remote access



VPN / IPsec

Network-layer tunneling



MACsec

Link-layer encryption



DNS Security

DNSSEC & DNS spoofing



IDS / IPS

Detection vs. prevention

WRAP-UP

The Toolkit Travels With You

Threat addressed · Trust model · Residual risk — this works for any mechanism, including ones that don't exist yet.

Exercises

1 TLS Handshake Trace

Trace a real TLS handshake and an encrypted data segment. Identify how the session is established — cipher suite, certificate, key exchange — and confirm what an observer can and cannot see.

2 Port Scan & Firewall

Your VM will be subjected to a port scan. Determine what it reveals about your exposed services, then configure the firewall to reduce what's reachable from outside.



Three Tools for Network Analysis

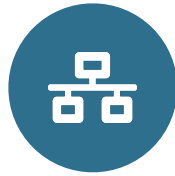


tcpdump

Command-line packet capture

Captures and displays live network traffic on any interface. Uses Berkeley Packet Filter (BPF) expressions to focus on the traffic you want — by host, port, or protocol. Available on virtually every Linux and macOS system; no GUI required.

tcpdump.org



Wireshark

GUI packet analyzer

Captures live traffic or opens saved .pcap files for offline analysis. Deep protocol dissection, flexible display filters, and TLS session decryption (with session keys). The standard tool for interactive, visual packet analysis.

wireshark.org/docs



nmap

Network mapper and port scanner

Discovers hosts and open ports; identifies running services and OS versions. Indispensable for understanding your own attack surface — and for seeing exactly what an attacker learns during reconnaissance.

nmap.org — [Reference Guide](#)