TBI-603

Network Security & Cyber Law

Unit Notes — Written for Listening

Unit One

Introduction to Network Security

Network security is the practice of protecting computers, networks, and data from unauthorized access, damage, or disruption. At its core, every security decision revolves around three fundamental goals — protecting the **confidentiality** of information, preserving its **integrity**, and guaranteeing its **availability**. Together, these three form what security professionals call the CIA triad. A security attack is any action that threatens one or more of these goals.

Security Attacks — Passive and Active

Security attacks fall into two broad categories. The first is the **passive attack**. In a passive attack, the adversary observes communication without altering anything. The data flows normally — the attacker simply listens. This is precisely what makes passive attacks so dangerous: they leave no visible trace. Two forms of passive attack exist. The first is the **release of message contents**, where the attacker reads confidential communication — an email, a file transfer, a private message. The second is **traffic analysis**, where even encrypted messages can reveal patterns. The attacker studies who is communicating, how often, and how much data is being exchanged. That alone can be valuable intelligence.

The second category is the **active attack**. Here the attacker does not merely watch — they intervene. Active attacks modify, interrupt, or fabricate data. They are more disruptive and generally easier to detect. Four types of active attacks are important to know. A **masquerade attack** occurs when the attacker impersonates a legitimate user, often using stolen credentials to gain unauthorized access. A **replay attack** involves capturing a valid message and retransmitting it later to deceive the receiving system. **Modification of messages** means the attacker intercepts and alters data in transit — for example, changing a payment amount in a bank transaction. Finally, a **Denial of Service attack** floods a system with so many requests that it becomes unable to serve legitimate

users.

> *To summarize the distinction: passive attacks threaten confidentiality and are hard to detect because nothing changes. Active attacks threaten integrity and availability and are more visible because they disrupt normal operation.*

The ISO Security Architecture

To bring order to all of this, the International Organization for Standardization developed a security architecture that organizes the field into three interconnected layers: the attacks themselves, the services designed to counter them, and the mechanisms that implement those services.

The six core **security services** are as follows. **Authentication** confirms the identity of the party you are communicating with. **Access control** determines who is permitted to use which resources. **Data confidentiality** ensures that only authorized parties can read the information being transmitted. **Data integrity** guarantees that the data has not been tampered with during transit. **Non-repudiation** prevents a party from later denying that they sent or received a message. And **availability** ensures that systems and services remain accessible when needed.

The **security mechanisms** that implement these services include encryption, which transforms readable data into an unreadable form using algorithms such as AES. Digital signatures verify the authenticity and integrity of messages. Access control systems enforce permissions and privileges. And authentication protocols, such as Kerberos, verify the identity of users before granting access.

Kerberos — The Authentication Protocol

Kerberos is a network authentication protocol developed at the Massachusetts Institute of Technology. It solves a fundamental problem in network security: how do you prove who you are to multiple services without repeatedly sending your password across the network? The answer lies in tickets and session keys.

Kerberos involves four components: the **Client** who wants access, the **Authentication Server** which verifies identity, the **Ticket Granting Server** which issues access passes, and the **Service Server** which provides the requested service.

The process unfolds in stages. The client sends a request to the Authentication Server. The server verifies the user's credentials and responds with a **Ticket Granting Ticket** — a secure, time-limited pass — along with a session key encrypted with the user's password. The client then takes this ticket to the Ticket Granting Server and requests access to a specific service. The Ticket Granting Server issues a **Service Ticket**, which the client presents to the Service Server to gain access. Throughout this entire exchange, the user's password never travels across the network. Only encrypted tickets do. This is the elegance of Kerberos.

> *Kerberos advantages: mutual authentication is supported, passwords are never exposed on the network, and replay attacks are resisted. Its limitations: it requires all clocks to be synchronized, and it introduces a central point of failure in the authentication servers.*

X.509 and Digital Certificates

X.509 is a standard for digital certificates used within Public Key Infrastructure. A digital certificate is essentially an electronic identity document — it binds a public key to the identity of an individual, organization, or server, and is issued by a trusted entity called a **Certificate Authority**.

Each certificate contains the version number, a serial number, the name of the issuing authority, the name of the subject, the subject's public key, an expiry date, and the Certificate Authority's digital signature. When you connect to an HTTPS website, your browser verifies the site's X.509 certificate. If the Certificate Authority is trusted and the certificate is valid, a secure connection is established. X.509 is the foundation of SSL, TLS, and VPNs.

Diffie-Hellman Key Exchange — Numerical

Given the values p equals 23, g equals 5, Alice's private key a equals 6, and Bob's private key b equals 15, the Diffie-Hellman exchange proceeds as follows.

Alice's public key: A = 5 to the power 6, mod 23 = 15625 mod 23 = 8

Bob's public key: B = 5 to the power 15, mod 23 = 19

Shared secret key: K = 19 to the power 6, mod 23 = 2

Both Alice and Bob independently arrive at the value 2, which becomes their shared secret. Neither party ever transmitted this secret — they derived it mathematically. This is the power of Diffie-Hellman.

Unit Two

Application and Transport Layer Security

This unit moves up the network stack to examine how security is applied at the application level — particularly in email communication — and at the transport level, where the protocols SSL and TLS provide the encrypted tunnels through which most of today's internet traffic flows.

Email Security

Email remains the most frequently exploited communication channel in organizational attacks. Without security measures in place, email is vulnerable to a range of threats: unauthorized interception, sender impersonation, phishing campaigns, malware-laden attachments, and deliberate alteration of message content during transmission.

Securing email requires addressing five needs simultaneously: confidentiality, integrity, authentication, non-repudiation, and protection against malware. Two widely used standards achieve this.

PGP — Pretty Good Privacy — operates on a decentralized model called the web of trust. Users vouch for each other's identities by signing one another's public keys. PGP is flexible, does not require a certificate authority, and is popular among individuals and technical users. However, it is less standardized and can be harder to manage at scale.

S/MIME — Secure Multipurpose Internet Mail Extensions — uses a centralized Certificate Authority model. Every user must obtain a certificate from a trusted authority. This makes S/MIME more structured and verifiable, and it is the preferred standard in corporate and government environments. Both PGP and S/MIME use public key encryption and digital signatures, but they differ in trust model, certificate requirements, and organizational applicability.

The SET Protocol

The Secure Electronic Transaction protocol was jointly developed by Visa and MasterCard to bring security to online credit card payments. Its three core objectives are to protect the confidentiality of payment details, to authenticate all parties involved in the transaction, and to ensure the integrity of every transaction record. SET uses digital certificates for the customer, the merchant, and the payment gateway — each issued by a Certificate Authority. While SET is not widely deployed today, it remains an important conceptual framework for secure payment systems.

SSL and TLS — Securing the Transport Layer

SSL, the Secure Socket Layer, was the original protocol designed to encrypt communication between a browser and a web server. TLS, the Transport Layer Security protocol, is its successor — more secure, more efficient, and the standard in use today. When you see HTTPS in your browser's address bar, TLS is working behind the scenes.

A TLS session is established through a carefully choreographed sequence. It begins with the **Client Hello**, in which the browser announces its supported TLS version, the list of cipher suites it can work with, and a randomly generated number. The server responds with a **Server Hello**, selecting the cipher suite to be used, sending its digital certificate, and contributing its own random number. The client then verifies the server's certificate against a trusted Certificate Authority. Once verified, both parties engage in a key exchange to derive a shared **session key**. From that point forward, all communication is encrypted symmetrically using that session key.

> *The key difference between SSL and TLS: SSL was developed by Netscape and is now considered insecure. TLS was standardized by the IETF and uses stronger cryptographic algorithms, better protection against known attacks, and offers improved performance.*

The Handshake and Record Protocols

TLS is internally divided into two sub-protocols, each with a distinct role. The **Handshake Protocol** operates before any application data is exchanged. Its job is to authenticate the parties, negotiate the cipher suite, and establish the session keys. Think of it as the formal introduction that must happen before the conversation can begin.

Once the handshake is complete, the **Record Protocol** takes over for the remainder of the session. It receives application data, breaks it into manageable fragments called records, optionally compresses them, encrypts each record using the negotiated session key, and attaches a Message Authentication Code to verify that the record has not been tampered with in transit.

Wireless TLS — WTLS

Wireless Transport Layer Security extends the principles of TLS to wireless and mobile environments. It is part of the Wireless Application Protocol architecture and is designed for networks with limited bandwidth and devices with constrained processing power. WTLS provides the same three security guarantees — confidentiality, authentication, and integrity — but in a lighter, more efficient form suitable for mobile banking, wireless browsing, and mobile commerce.

TLS Record Calculation — Numerical

A 3000-byte message is to be transmitted using TLS. Each TLS record can carry a maximum of 600 bytes of data, and each record adds 24 bytes of overhead.

Number of records = 3000 ÷ 600 = 5 records

Total overhead = 5 × 24 = 120 bytes

Total transmitted = 3000 + 120 = 3120 bytes

Unit Three

IP Security and System Security

This unit descends to the network layer — the level at which raw IP packets travel between systems — and examines how security can be applied directly to those packets. It also looks inward at the operating system and software themselves, asking how we protect the machines that run our applications.

IPsec — Authentication Header and ESP

IPsec is a suite of protocols that provides security at the IP layer. It operates through two distinct mechanisms, each serving a different purpose.

The **Authentication Header**, or AH, provides three guarantees: it authenticates the origin of a packet, verifies that the packet has not been altered in transit, and protects against replay attacks by including a sequence number in each packet. What AH does not provide is encryption. The contents of the packet remain visible. AH is the right choice when the integrity and authenticity of a packet matter, but confidentiality is not required.

The **Encapsulating Security Payload**, or ESP, goes further. In addition to authentication, integrity, and anti-replay protection, ESP encrypts the packet payload, ensuring that its contents cannot be read by anyone who intercepts it. ESP is more widely used than AH precisely because it offers the complete security package: confidentiality, authentication, and integrity in a single protocol.

Transport Mode and Tunnel Mode

IPsec can operate in one of two modes, and the difference between them is significant.

In **transport mode**, only the payload — the data portion — of the IP packet is protected. The original IP header, which contains the source and destination addresses, remains unchanged and visible. This mode introduces less overhead

and is used for direct communication between two hosts.

In **tunnel mode**, the entire original IP packet — header and payload together — is encapsulated within a new IP packet. A new outer IP header is added, completely concealing the original addresses. This is the mode used by Virtual Private Networks. When a company's branch office communicates with its headquarters through a VPN, tunnel mode ensures that the internal IP structure remains hidden from anyone observing the traffic.

Intrusion Detection Systems

An intrusion is any unauthorized attempt to access, damage, or disrupt a computer system. An Intrusion Detection System, or IDS, monitors network traffic and system activity continuously, looking for behavior that matches known attack patterns or deviates significantly from established norms. When a threat is detected, the IDS generates an alert for the security team to investigate.

There are two types. A **Host-based IDS** is installed on an individual machine and monitors that machine's logs, file system changes, and system calls. A **Network-based IDS** sits on the network itself and monitors all traffic passing through, examining packet headers and payloads for signatures of known attacks.

Firewalls

A firewall is a security boundary — hardware, software, or both — that controls what traffic is allowed to enter or leave a network. Three principles govern sound firewall design: all traffic between networks must pass through the firewall, only traffic explicitly authorized by the security policy is permitted to pass, and the firewall itself must be hardened against attack.

Firewalls come in several forms. **Packet filtering** firewalls examine each packet individually against a set of rules. **Stateful inspection** firewalls track the state of active connections and permit only packets that belong to established sessions. **Proxy firewalls** act as intermediaries, making requests on behalf of clients and

shielding internal systems from direct exposure. **Application firewalls** operate at the highest layer and can inspect the content of application-level protocols.

Trusted Systems and Program Security

A trusted system is an operating environment that has been specifically designed and verified to enforce a defined security policy reliably and correctly. Features of trusted systems include mandatory access control, robust user authentication, comprehensive audit trails that log all security-relevant events, data integrity verification mechanisms, and continuous security monitoring. Trusted systems are essential in multi-user environments where different users require different levels of access to sensitive resources.

Program security addresses the vulnerability of software itself. Common threats include buffer overflow exploits, malware injection, unauthorized code modification, and software bugs. Defenses include adopting secure coding practices from the development stage, validating all user input before processing, enforcing strict access controls on program resources, deploying antivirus and anti-malware tools, and maintaining a rigorous patch management process.

Viruses, Worms, and Malicious Code

Malicious software takes many forms, and the distinctions between them matter.

A **virus** is code that attaches itself to a legitimate host file. It lies dormant until the infected file is executed, at which point it activates and may spread to other files. A virus requires a host and requires a human action — running a file — to propagate.

A **worm** requires neither a host file nor human action. It is self-replicating and spreads autonomously through network connections, consuming bandwidth and compromising systems as it goes.

A **Trojan horse** presents itself as legitimate, useful software while concealing a malicious payload. It may create backdoors, exfiltrate data, or disable security

systems. A **spyware** program operates silently in the background, collecting keystrokes, browsing history, and credentials. **Ransomware** encrypts the victim's files and demands a ransom payment in exchange for the decryption key.

Unit Four

Cyber Law and Digital Governance

This unit steps outside the technical domain and into the legal one. Technology alone cannot create a trustworthy digital environment — laws must define what is permissible, establish consequences for violations, and provide legal standing to digital transactions and documents.

Cyber Law, Cyber Crime, and Cyber Criminals

Cyber law is the body of legislation that governs activities in cyberspace — internet usage, electronic communication, digital commerce, and the handling of electronic data. It provides the legal infrastructure within which the digital economy operates.

Cyber crime refers to any illegal act carried out using a computer, a network, or the internet as the instrument or target. Examples include unauthorized access to computer systems, identity theft, online financial fraud, phishing, cyber stalking, and the distribution of malware.

A **cyber criminal** is any individual or organized group that commits such acts — including hackers who exploit system vulnerabilities, fraudsters who conduct online scams, malware developers who create and distribute harmful software, and cyber terrorists who attack critical infrastructure.

The Information Technology Act, 2000

The Information Technology Act of 2000 is India's foundational legislation for the digital domain. Its genesis lies in the rapid growth of electronic commerce in the late 1990s and the corresponding need for a legal framework to govern it. The Act was directly influenced by the UNCITRAL Model Law on Electronic Commerce, a framework developed by the United Nations to promote consistent legal treatment of e-commerce across jurisdictions.

The IT Act serves five principal objectives. First, it grants legal recognition to electronic records, placing them on equal legal footing with paper documents. Second, it legally recognizes digital signatures, making digitally signed documents enforceable in law. Third, it promotes and facilitates electronic commerce by providing a legal environment in which online transactions can be conducted with confidence. Fourth, it defines cyber offences and prescribes penalties for unauthorized access, data theft, and other digital crimes. Fifth, it supports e-governance by enabling government agencies to deliver services and maintain records in electronic form.

The scope of the Act extends to electronic records of all kinds, digital signatures, electronic contracts, online communications, and cyber offences — covering virtually the entire spectrum of digital activity.

Legal Recognition of Electronic Records

An electronic record is any information that is generated, communicated, received, or stored in electronic form — this includes emails, digital contracts, online forms, electronic databases, and digitally stored documents.

Before the IT Act, Indian law recognized only physical documents as legally valid. The Act changed this fundamentally: an electronic record now carries the same legal weight as its paper equivalent. A contract concluded over email is as binding as one signed on paper. This recognition removed a critical barrier to digital commerce and e-governance, enabling businesses and government agencies alike to operate in a fully digital environment.

Legal Recognition of Digital Signatures

A digital signature is a cryptographic mechanism that authenticates the identity of the signer and verifies that a document has not been altered since it was signed. It is generated using the signer's private key and can be verified by anyone with access to the corresponding public key.

The IT Act grants digital signatures full legal validity. A document bearing a valid digital signature is treated under Indian law as equivalent to a handwritten signature. Digital signatures provide four critical assurances: **authentication** — confirming the signer's identity, **integrity** — ensuring the document has not been modified after signing, **non-repudiation** — preventing the signer from later denying that they signed the document, and **security** — protecting the authenticity of the communication.

In practice, government agencies use digital signatures for income tax e-filing, public procurement through e-tendering, the issuance of official certificates and licenses, and passport application processing — among many other administrative functions.

Cyber Law in E-Commerce and E-Governance

In the context of **electronic commerce**, cyber law performs several essential functions. It provides legal recognition to contracts concluded online, ensuring they are enforceable. It protects consumers by defining legal remedies for fraud and deceptive practices. It establishes the legal framework for secure digital payments and mandates the protection of customer data.

In the context of **electronic governance**, cyber law enables the delivery of public services through digital platforms, reduces the cost and inefficiency of paper-based administration, ensures the authenticity of government communications through digital signatures, and promotes transparency and accountability in public institutions.

Scanning Techniques

Before launching an attack, adversaries typically conduct reconnaissance to understand the target network. Scanning is the systematic process of identifying active hosts, open ports, running services, and potential vulnerabilities.

Ping sweeping sends ICMP Echo Request messages to a range of IP addresses. Systems that respond are confirmed as active. **Port scanning** probes

a target system's TCP and UDP ports to determine which are open and what services are listening. Common variants include the TCP connect scan, the SYN scan — which is stealthier because it never completes the handshake — and the UDP scan.

ICMP scanning uses a variety of ICMP message types beyond the simple echo request to gather network topology information. **Fingerprinting** identifies the operating system and software versions running on a target. Active fingerprinting sends specially crafted packets and analyzes the responses. Passive fingerprinting observes existing traffic without introducing any new packets, making it much harder to detect.

Unit Five

Buffer Overflow and DoS Attacks

This final unit examines the attack techniques themselves — the specific methods adversaries use to compromise software, disrupt services, and exploit the behavior of networks. Understanding these attacks in depth is essential not only for defending against them, but for answering examination questions with precision.

Buffer Overflow Attacks

Every running program allocates a fixed region of memory called a buffer to hold incoming data. A buffer overflow occurs when more data is written into that buffer than it was designed to hold. The excess data does not simply disappear — it spills into the adjacent memory space, potentially overwriting data or instructions that the program depends on. Attackers deliberately engineer this overflow to inject malicious instructions into memory and redirect the program's execution to those instructions.

There are three distinct forms of this attack. A **stack overflow** targets the call stack — the region of memory that manages function calls. Each time a function is called, a return address is pushed onto the stack, telling the program where to resume execution when the function completes. If an attacker can overflow a stack buffer, they can overwrite this return address and redirect execution to any location in memory — including injected malicious code.

A **heap overflow** targets the heap — the region of memory that programs use for dynamic allocation at runtime. Overflowing a heap buffer corrupts the data structures that manage the heap itself, leading to security vulnerabilities that can be exploited to execute arbitrary code or crash the system.

An **integer overflow** is a subtler form of the problem. When an arithmetic operation produces a result larger than the maximum value the variable can

store, the value wraps around to an unexpectedly small or negative number. This can trick a program into allocating an undersized buffer, creating the conditions for a subsequent overflow.

Internal Attacks

Not all threats come from outside the organization. Internal attacks are carried out by individuals who already have authorized access — employees, contractors, or other insiders who misuse their privileges.

Email abuse involves the misuse of organizational email systems to distribute spam, leak confidential data externally, or deliver phishing messages to colleagues. **Mobile phone misuse** refers to the use of work devices to exfiltrate sensitive data or inadvertently introduce malware by connecting to insecure external networks. **Instant messenger threats** arise when employees use chat applications to share malware links or leak sensitive business information to unauthorized parties.

FTP upload abuse occurs when individuals upload unauthorized, malicious, or sensitive files to organizational file servers, either to distribute them externally or to compromise internal systems. **Shoulder surfing** is the simplest form of internal attack — physically observing another person entering a password, PIN, or other sensitive information. Effective countermeasures include security awareness training, strict access control policies, activity monitoring, and clearly communicated acceptable use policies.

Denial of Service and Distributed Denial of Service

A **Denial of Service attack** is an attempt to make a system, service, or network resource unavailable to its intended users by overwhelming it with a flood of illegitimate requests. A single attacking machine directs a sustained volume of traffic or requests at the target until it can no longer respond to legitimate users.

A **Distributed Denial of Service attack** scales this concept dramatically. The attacker first compromises a large number of machines — forming what is called

a **botnet** — and then commands all of them to attack a single target simultaneously. The sheer volume of traffic generated by thousands of machines makes DDoS attacks far more difficult to detect and mitigate than simple DoS attacks originating from a single source.

Four DoS Attack Variants You Must Know

The **Ping of Death** exploits a limitation in the ICMP protocol. The maximum legal size of an IP packet is 65,535 bytes. The Ping of Death sends ICMP packets that exceed this limit. When the target system attempts to reassemble the oversized packet, it crashes or becomes unstable.

SYN Flooding exploits the TCP three-way handshake. In normal operation, a client sends a SYN packet, the server responds with SYN-ACK, and the client completes the handshake with an ACK. In a SYN flood, the attacker sends thousands of SYN packets but never sends the final ACK. The server holds each half-open connection in memory, waiting for the ACK that never arrives. Eventually the connection table is full, and the server can no longer accept legitimate connections.

The **Smurf Attack** uses amplification. The attacker spoofs the victim's IP address and sends ICMP Echo Request packets to a network's broadcast address. Every device on that network receives the request and sends an ICMP Echo Reply — all directed at the victim's IP address. A single request can generate hundreds of replies, amplifying the attack traffic significantly.

UDP Flooding sends an overwhelming volume of UDP packets to random ports on the target. For each packet, the target system checks whether any application is listening on that port, finds none, and sends back an ICMP Destination Unreachable message. Processing this flood of packets and generating all those responses exhausts the target's bandwidth and CPU resources.

IP Address Hiding and Tracing

Attackers routinely attempt to conceal their true IP address to avoid attribution and prosecution. The common techniques are: routing traffic through a **proxy server**, which acts as an intermediary; using a **VPN**, which replaces the user's real IP with the VPN server's address; routing through the **TOR network**, which passes traffic through multiple relays, each knowing only the previous and next hop; and **IP spoofing**, which forges the source address field in packet headers.

Investigators employ several techniques to trace attack origins. **Traceroute** reveals the path a packet takes from source to destination, identifying intermediate routers. **Packet analysis** examines captured network traffic for identifying characteristics. **Log analysis** reviews server, router, and firewall logs for patterns that reveal the true origin. **Network monitoring** tracks traffic behavior over time, identifying anomalies that point to an attacker's infrastructure.

Mitigation — The Complete Picture

No single measure is sufficient against the full range of attacks covered in this unit. Effective defense requires layering: firewalls filter malicious traffic at the perimeter. Intrusion detection and prevention systems identify threats in real time and block them before damage occurs. Traffic filtering and rate limiting reduce the impact of flooding attacks. Load balancing distributes traffic across multiple servers, preventing any single point from being overwhelmed. Regular patching closes the software vulnerabilities that buffer overflow attacks depend on. Multi-factor authentication limits the damage an insider can cause with stolen credentials. Security awareness training reduces the likelihood that employees will fall victim to phishing or commit intentional abuse. And comprehensive backup systems ensure that when an attack succeeds, recovery is possible.