UNIT-IV

Network Security: TLS, SSL, SSH & Wireless Security

MODULE 4

Agenda — Module 4

- Transport Layer Security (TLS)
- Web Security Considerations
- SSL vs TLS Comparison
- SSH — Secure Shell Protocol
- Wireless Security (WEP, WPA, WPA2, WPA3)
- Secure Socket Layer (SSL) Deep Dive
- HTTP and HTTPS
- Mobile Device Security
- IEEE 802.11 Architecture & Services

Transport Layer Security

TLS — Securing the Transport Layer

What is TLS?

- TLS (Transport Layer Security) provides security at the transport layer
- Derived from SSL (Secure Socket Layer)
- Ensures no third party can eavesdrop or tamper with messages
- Successor to SSL — more secure and widely used today
- Protects data in transit between client and server

Benefits of TLS

- **Encryption — Secures transmitted data using strong encryption**
- **Interoperability — Works with most browsers, OS, and web servers**
- **Algorithm Flexibility — Supports multiple auth, encryption & hashing algorithms**
- **Ease of Deployment — Deployable on Windows Server and major platforms**
- **Ease of Use — Operates mostly invisible beneath the application layer**

How TLS Works — Handshake Process

- Client connects to server via TCP
- Client sends: SSL/TLS version, supported cipher suites & compression methods
- Server selects highest mutual TLS version and a cipher suite
- Server sends its certificate (must be trusted by client or a trusted CA)
- Client verifies certificate — confirms server identity (prevents MITM)
- Key exchange occurs (public key / PreMasterSecret)
- Both parties compute symmetric encryption key
- Handshake complete — secure communication begins

TLS at a Glance

3

TLS Versions

1.0, 1.2, 1.3 (current)

256

Bit Encryption

AES-256 in TLS 1.3

0-RTT

Handshake

Zero round-trip in TLS 1.3

Web Security Considerations

Protecting Websites & Web Applications

Why Web Security Matters

- Websites are constantly prone to security risks
- Cybercrime can hack your website for malicious assaults
- Hackers may steal customer data (credit cards, passwords)
- Malware can be installed on visitors' computers
- Business reputation and customer trust can be destroyed
- Illegal content can be propagated to your users

Web Security Best Practices

- **Updated Software — Keep all software current to patch vulnerabilities**
- **SQL Injection Prevention — Sanitize inputs; never trust user data**
- **Cross-Site Scripting (XSS) — Encode/strip HTML in form submissions**
- **Error Messages — Do not reveal which field (username/password) is incorrect**
- **Data Validation — Validate on both server-side AND client-side**
- **Passwords — Enforce min 8 chars, uppercase, lowercase & special characters**
- **File Uploads — Restrict executable file types from being uploaded**
- **SSL — Use SSL when passing personal data between site and server**

Top Web Security Threats

Injection & Scripting

- Cross-Site Scripting (XSS)
- SQL Injection
- Code Injection

Malware & Attacks

- Phishing
- Ransomware
- Viruses and Worms
- Spyware
- Denial of Service (DoS)

SQL Injection — Deep Dive

- An attempt by hackers to manipulate your database through form inputs
- Attackers insert rogue SQL code into queries via user input fields
- Can be used to: change tables, extract data, delete records
  - Prevention: Use parameterized queries / prepared statements
  - Prevention: Input validation and sanitization
  - Prevention: Principle of least privilege for DB accounts
  - Prevention: Use ORM frameworks that handle escaping

Cross-Site Scripting (XSS)

- Allows attackers to inject client-side scripts into web pages
- Script runs in the victim's browser with the trust of the site
- Can steal cookies, session tokens, redirect users
  - Types: Stored XSS, Reflected XSS, DOM-based XSS
  - Prevention: Encode/escape all user-supplied data in output
  - Prevention: Use Content Security Policy (CSP) headers
  - Prevention: Validate and sanitize all form data inputs

SSL vs TLS

Comparing Secure Socket Layer and Transport Layer Security

SSL vs TLS — Key Differences (1/2)

Feature	SSL	TLS
Full Name	Secure Socket Layer	Transport Layer Security
Algorithm	Supports Fortezza algorithm	Does NOT support Fortezza
Version	SSL 3.0	TLS 1.0+
Master Secret	Uses Message Digest	Uses Pseudo-random function
MAC Protocol	MAC (Message Auth Code)	HMAC (Hashed MAC)

SSL vs TLS — Key Differences (2/2)

Feature	SSL	TLS
Complexity	More complex	Simpler
Security	Less secure	High security
Performance	Slower, less reliable	Faster, highly reliable
Status	Deprecated	Widely used (standard)
Connection	Explicit (port-based)	Implicit (protocol-based)

SSH — Secure Shell Protocol

Cryptographic Network Protocol for Secure Communication

What is SSH?

- SSH = Secure Shell (or Secure Socket Shell)
- Cryptographic network protocol for secure communication
- Allows two computers to communicate over an insecure network
- Used to: log in to remote servers, execute commands, transfer data
- Developed by SSH Communications Security Ltd
- Replaced insecure protocols: Telnet, rlogin, rsh, FTP
- Protects against: DNS spoofing, IP source routing, IP spoofing

SSH — Key Use Cases

- Secure access for users and automated processes
- Secure file transfer from one system to another (SFTP/SCP)
- Issue remote commands to users and systems
- Manage network infrastructure and critical system components
- Log into remote shell, replacing Telnet and rlogin
- Combine with rsync for secure backup, copy, and mirroring
- Port forwarding through encrypted tunnels
- Set up automatic login via public key authentication (OpenSSH)
- Browse web securely through encrypted SOCKS proxy

How SSH Works

- SSH operates on a client-server model
  - SSH Client — End where the session is displayed
  - SSH Server — End where session executes
- Provides strong password authentication
- Encrypted communication using public key cryptography
- Uses asymmetric keys for handshake, symmetric for session
- Port 22 is the standard SSH port
- Session encrypted end-to-end throughout communication

Wireless Security

WEP, WPA, WPA2, WPA3 — Evolution of Wi-Fi Security

Wireless Security — Overview

- Securing wireless networks from malicious attempts and unauthorized access
- **Hardware-based — Routers/switches with built-in encryption**
- **Wireless IDS/IPS — Detect, alert, and prevent unauthorized access**
- **Wireless Security Algorithms — WEP, WPA, WPA2, WPA3**
- Even if data is compromised, encryption prevents viewing content
- Network admin receives alarms in case of any security breach

WEP — Wired Equivalent Privacy (1999)

- Oldest wireless security algorithm (introduced in 1999)
- Uses Initialization Vector (IV) method for encryption
- Originally limited to 64-bit encryption due to U.S. export laws
- Later expanded to 128-bit and 256-bit WEP (128-bit became standard)
- Critical flaw: Key-scheduling vulnerability discovered
- WEP key can be cracked in minutes using automated tools
- NOW CONSIDERED INSECURE — do not use unless no alternative exists

WPA — Wi-Fi Protected Access (2003)

- Developed by Wi-Fi Alliance to replace WEP vulnerabilities
- Officially adopted in 2003, one year before WEP retirement
- Uses TKIP (Temporal Key Integrity Protocol) for data encryption
- Introduced 802.1x authentication for improved user authentication
- Most common config: WPA-PSK (Pre-Shared Key)
- Uses 256-bit keys — significant improvement over WEP's 64/128-bit
- Still has vulnerabilities; superseded by WPA2

WPA2 vs WPA3

WPA2 (2006)

- Official standard since 2006
- Uses AES (Advanced Encryption Standard)
- Uses CCMP replacing TKIP
- Counter Cipher Mode — Block Chaining Message Authentication
- Still widely deployed today
- Vulnerable to KRACK attacks

WPA3 (Latest)

- Latest Wi-Fi Alliance iteration
- 384-bit Hashed Message Auth Mode
- 256-bit GCMP-256 encryption
- 256-bit Broadcast/Multicast Integrity
- Perfect Forward Secrecy support
- Personal & enterprise security support

SSL — Secure Socket Layer

Protocols, Stack & Handshake Process

SSL Protocol Stack

- **SSL Record Protocol**
  - Provides Confidentiality and Message Integrity
- **SSL Handshake Protocol**
  - Establishes sessions; authenticates client and server
- **Change-Cipher Spec Protocol**
  - Signals transition to newly negotiated cipher suite
- **Alert Protocol**
  - Conveys SSL-related alerts and errors between peers

SSL Record Protocol — Detail

- Divides application data into fragments
- Compresses each fragment
- Appends MAC (Message Authentication Code)
  - MAC generated using SHA (Secure Hash Protocol) or MD5
- Encrypts the compressed, MAC-appended data
- Appends SSL header to the encrypted data
- Two services provided: Confidentiality and Message Integrity

SSL Handshake Protocol — 4 Phases

- **Phase 1 — Hello Exchange**
  - Client and Server exchange hello packets; share cipher suite, protocol version, IP session
- **Phase 2 — Server Authentication**
  - Server sends certificate and Server-key-exchange; ends with Server-hello-end
- **Phase 3 — Client Authentication**
  - Client replies with its certificate and Client-exchange-key
- **Phase 4 — Cipher Change & Finish**
  - Change-cipher suite occurs; handshake protocol ends

HTTP and HTTPS

Hypertext Transfer Protocol & Secure Communication

HTTP — Hypertext Transfer Protocol

- Application-level protocol for distributed, collaborative hypermedia systems
- Foundation for data communication on the World Wide Web since 1990
- Generic and stateless protocol — each request is independent
- TCP/IP based — delivers HTML, images, query results over the web
- Default port: TCP 80 (HTTP) | TCP 443 (HTTPS)
- Standardizes how clients request data and servers respond

HTTP — Three Basic Features

- **HTTP is Connectionless**
  - Browser makes request; server responds; connection is dropped. Next request starts fresh.
- **HTTP is Media Independent**
  - Any type of data can be sent as long as both sides know how to handle it (via MIME types)
- **HTTP is Stateless**
  - Server and client know each other only during current request; no memory between requests

HTTP Header Types

- **General Headers — Applicable to both request and response messages**
- **Client Request Headers — Applicable only for request messages**
- **Server Response Headers — Applicable only for response messages**
- **Entity Headers — Define meta information about the entity-body**
- Content-Type and Content-Length specify the nature of the message body
- HTTP/1.0 uses new connection per request; HTTP/1.1 allows reuse

HTTP Security Considerations

- HTTP communications over the internet carry inherent security risks
- **Personal Information Leakage**
  - HTTP clients may transmit names, location, email, passwords, encryption keys
  - Prevention: Never send sensitive data over plain HTTP
- Always use HTTPS (HTTP over TLS/SSL) for secure web communication
- HTTPS encrypts all data in transit between browser and server
- HTTPS uses port 443 by default

Mobile Device Security

Protecting Laptops, Smartphones & Tablets

Mobile Device Security — Overview

- Protects sensitive information stored and transmitted by portable devices
- Covers: laptops, smartphones, tablets, wearables, portable devices
- Core goal: prevent unauthorized users from accessing enterprise network
- Over half of business PCs are now mobile — creating new challenges
  - Threats include: malicious apps, phishing, data leakage, spyware, rogue Wi-Fi
  - Also: lost or stolen devices exposing company data

Benefits of Mobile Device Security

- Regulatory compliance enforcement
- Security policy enforcement across all devices
- Support for BYOD (Bring Your Own Device)
- Remote control of device updates
- Application control and management
- Automated device registration
- Data backup and recovery
- Protection from malicious outsiders accessing sensitive company data

Mobile Security Best Practices

- **Clear Policies — Define allowed devices, OS levels, access rules**
- **Strong Passwords — Min 8 chars, unique per account, updated regularly**
- **Biometrics — Face, fingerprint, voice, or iris recognition**
- **Avoid Public Wi-Fi — Educate employees; prohibit open networks**
- **App Control — Ban or restrict unapproved app downloads**
- **Remote Wipe — Enable IT to remotely wipe lost/stolen devices**

IEEE 802.11 Architecture

Wi-Fi Standards, Frame Format & RSN Security

IEEE 802.11 — Architecture & RSN Services

- **Stations (STA) — All devices connected to wireless LAN**
  - WAP (Wireless Access Points) — Base stations forming the network
  - Clients — Workstations, laptops, smartphones, printers
- **BSS (Basic Service Set) — Group of stations at physical layer**
- **ESS (Extended Service Set) — All connected BSS combined**
- **RSN Security Services (IEEE 802.11i)**
  - Authentication — Mutual auth + temporary key generation
  - Access Control — Routes messages; works with auth protocols
  - Privacy with Message Integrity — MAC-level encryption + integrity code