UNIT–1: Fundamentals of Web Application Security

1. History of Software Security

- Early software focused mainly on functionality, not security.
- Rise of internet → increase in cyber-attacks (worms, viruses, SQL injection, etc.).
- Security evolved from reactive patches to proactive secure coding.

2. Recognizing Web Application Security Threats

- Threats include: malware, phishing, SQL injection, XSS, session hijacking, broken access control.
- Attackers exploit vulnerabilities in code, configuration, or user behavior.

3. Web Application Security

- Protecting web apps from attacks by securing code, server, and data.
- Includes: secure coding, patching, encryption, authentication, and access control.

4. Authentication and Authorization

- **Authentication**: Verifies user identity (passwords, OTPs, biometrics).
- **Authorization**: Determines what authenticated users can access (role-based access).

5. Secure Socket Layer (SSL) & Transport Layer Security (TLS)

- Encryption protocols that secure communication between client and server.
- Provide confidentiality, integrity, and authenticity using certificates.

6. Session Management

- Maintains user state after login using session IDs or cookies.
- Secure session handling prevents hijacking, fixation, and replay attacks.

7. Input Validation

- Ensures that user input is filtered and sanitized.
- Prevents major attacks like SQL Injection, XSS, Command Injection.

UNIT–2: Secure Development and Deployment

1. Web Application Security

- Developing apps with security principles from design to deployment.
- Includes secure coding, encryption, least privilege, and secure configurations.

2. Security Testing

- Evaluates application vulnerabilities using tests like SAST, DAST, fuzzing.
- Helps detect weaknesses before deployment.

3. Security Incident Response Planning

- A predefined plan for detecting, analyzing, and responding to attacks.
- Phases: preparation, detection, containment, eradication, recovery, lessons learned.

4. Microsoft Security Development Lifecycle (SDL)

- A secure development process with phases: requirements, design, implementation, testing, release, response.
- Focuses on threat modeling and secure coding guidelines.

5. OWASP CLASP (Comprehensive Lightweight Application Security Process)

- A set of security practices integrated into SDLC.
- Focuses on roles, responsibilities, and vulnerability prevention.

6. Software Assurance Maturity Model (SAMM)

- Framework to measure and improve software security practices.
- Four domains: Governance, Construction, Verification, Deployment.

UNIT–3: Secure API Development

1. API Security

- Protecting APIs from unauthorized access, data leakage, and abuse.

2. Session Cookies

- Store session IDs.
- Should be secure, HttpOnly, and have proper expiration.

3. Token-Based Authentication

- Uses tokens (JWT, OAuth tokens) instead of sessions.
- Easier for mobile and distributed systems.

4. Securing Natter APIs (REST APIs)

- Prevent API abuse and vulnerabilities using:
    - **Security controls** (input validation, authentication, access control).
    - **Rate limiting** to prevent DDoS/burst requests.
    - **Encryption** (HTTPS).
    - **Audit logging** for monitoring API access.

5. Securing Service-to-Service APIs

- **API Keys** for basic identification.
- **OAuth2** for delegated authorization between services.

6. Securing Microservice APIs

- **Service Mesh** (e.g., Istio) provides authentication, traffic control, encryption.
- Lock down network connections and restrict ports.
- Validate all incoming requests to microservices.

UNIT–4: Vulnerability Assessment & Penetration Testing

1. Vulnerability Assessment Lifecycle

- Steps: Asset identification → scanning → analysis → remediation → verification → reporting.

2. Vulnerability Assessment Tools

- **Cloud-based scanners** → scan cloud apps (e.g., Qualys).
- **Host-based scanners** → scan OS and installed software.
- **Network-based scanners** → scan network devices and ports.
- **Database-based scanners** → check for insecure DB configurations.

3. Types of Penetration Tests

- **External Testing** → simulates outside attacker.
- **Web Application Testing** → tests app-specific vulnerabilities.
- **Internal Penetration Testing** → simulates insider attacks.
- **SSID/Wireless Testing** → tests Wi-Fi networks.
- **Mobile Application Testing** → tests mobile apps for security flaws.

UNIT–5: Hacking Techniques and Tools

1. Social Engineering

- Manipulating people to reveal confidential information (phishing, pretexting).

2. Injection

- Attacks that insert malicious commands (SQL injection, command injection).

3. Cross-Site Scripting (XSS)

- Injecting malicious scripts into web pages viewed by users.
- Types: Stored, Reflected, DOM-based.

4. Broken Authentication & Session Management

- Weak login and session controls allow attackers to hijack accounts.

5. Cross-Site Request Forgery (CSRF)

- Forces a logged-in user to perform unwanted actions (like transactions).

6. Security Misconfiguration

- Incorrect server/application settings create vulnerabilities (open ports, default credentials).

7. Insecure Cryptographic Storage

- Storing sensitive data without proper encryption or hashing.

UNIT–6: Failure to Restrict URL Access & Tools

1. Failure to Restrict URL Access

- Occurs when users access unauthorized pages via direct URL access.
- Prevented using authorization checks on each page.

2. Tools

- **Comodo** → Web security, SSL, malware scanning.
- **OpenVAS** → Open-source vulnerability scanner.
- **Nexpose** → Vulnerability management and risk scoring.
- **Nikto** → Web server scanner for outdated software and config issues.
- **Burp Suite** → Web penetration testing tool for scanning and intercepting traffic.

✅ UNIT–1: Fundamentals of Web Application Security (10 MCQs)

1. Software security initially focused on which of the following?

A. Secure coding
B. Application functionality
C. Access control
D. Threat modeling
Answer: B

2. Which of the following is NOT a web application threat?

A. SQL Injection
B. Phishing
C. XSS
D. Cloud Storage
Answer: D

3. Authentication means:

A. Giving permissions
B. Identifying a user
C. Encrypting data
D. Managing sessions
Answer: B

4. SSL/TLS provides:

A. Only integrity
B. Only authentication
C. Encryption, integrity, authentication
D. Only authorization
Answer: C

5. Session management mainly uses:

A. Tokens
B. Cookies and session IDs
C. Digital signatures
D. Backups
Answer: B

6. Input validation prevents:

A. Brute-force attacks
B. SQL Injection and XSS
C. Server outages
D. Patch failures
Answer: B

7. Broken access control is related to:

A. Encryption failure
B. Unauthorized resource access
C. Password hashing
D. Logging errors
Answer: B

8. TLS is the updated version of:

A. SSH
B. SSL
C. HTTPS
D. HTTP
Answer: B

9. A web application is secure when:

A. It has a strong UI
B. It is tested and configured securely
C. It uses only HTML
D. It is deployed on cloud
Answer: B

10. Which ensures confidentiality on the web?

A. Session cookies
B. TLS encryption
C. Authorization
D. Caching
Answer: B

✅ UNIT–2: Secure Development and Deployment (10 MCQs)

1. Security testing is used to find:

A. UI issues
B. Functional errors
C. Vulnerabilities
D. Documentation errors
Answer: C

2. Incident response begins with:

A. Recovery
B. Containment
C. Preparation
D. Analysis
Answer: C

3. SDL model was developed by:

A. IBM
B. Google
C. Microsoft
D. Oracle
Answer: C

4. CLASP is primarily focused on:

A. Hardware testing
B. Lightweight security practices
C. Front-end design
D. User authentication
Answer: B

5. SAMM includes how many domains?

A. 3
B. 4
C. 5
D. 7
Answer: B

6. Threat modeling is done during which SDL phase?

A. Design
B. Deployment
C. Testing
D. Release
Answer: A

7. Verification in SAMM involves:

A. Code writing
B. Testing security
C. Deployment configuration
D. User training
Answer: B

8. Secure coding is part of:

A. Construction phase
B. Governance phase
C. Release phase
D. Monitoring phase
Answer: A

9. Security testing like SAST is done on:

A. Running application
B. Static code
C. Cloud services
D. Network devices
Answer: B

10. Deployment security includes:

A. UI color selection
B. Patch installation
C. Database normalization
D. Python compilation
Answer: B

✅ UNIT–3: Secure API Development (10 MCQs)

1. REST API communication uses:

A. SSH
B. HTTP/HTTPS
C. STP
D. SMTP
Answer: B

2. Session cookies must be:

A. Visible to JavaScript
B. HttpOnly and Secure
C. Unencrypted
D. Publicly accessible
Answer: B

3. Token-based authentication commonly uses:

A. CSS
B. JWT
C. SQL
D. JSON-only
Answer: B

4. Rate limiting prevents:

A. Data encryption
B. API abuse and DoS
C. HTTPS usage
D. Database indexing
Answer: B

5. Audit logging helps in:

A. Speeding API
B. Monitoring access
C. Reducing encryption
D. Increasing UI quality
Answer: B

6. API keys are used for:

A. User UI design
B. Simple service-to-service authentication
C. Database queries
D. Encryption
Answer: B

7. OAuth2 provides:

A. Session management
B. Authorization delegation
C. Token hashing
D. Password encryption
Answer: B

8. Service Mesh is used in:

A. Monolithic apps
B. Microservices
C. Local storage
D. Browser caching
Answer: B

9. Encryption of API traffic is done using:

A. TLS
B. JSON
C. XML
D. HTML
Answer: A

10. Securing incoming API requests includes:

A. Allowing all IPs
B. Validating tokens
C. Disabling authentication
D. Using plain HTTP
Answer: B

✅ UNIT–4: Vulnerability Assessment & Penetration Testing (10 MCQs)

1. First step in vulnerability assessment lifecycle:

A. Reporting
B. Analysis
C. Asset identification
D. Remediation
Answer: C

2. OpenVAS is an example of:

A. Host scanner
B. Network scanner
C. Wireless tool
D. Firewall
Answer: B

3. External penetration test simulates:

A. Insider
B. Outsider
C. Admin
D. System user
Answer: B

4. Web application penetration testing focuses on:

A. Cloud architecture
B. Application layer vulnerabilities
C. Physical devices
D. Password length
Answer: B

5. Wireless testing evaluates:

A. RAM size
B. Wi-Fi vulnerabilities
C. Browser cookies
D. SQL databases
Answer: B

6. Database vulnerability scanners detect:

A. Table joins
B. Misconfigurations
C. SSL certificates
D. Session cookies
Answer: B

7. Cloud-based scanners are used for:

A. Offline servers
B. Cloud-hosted apps
C. Desktop apps
D. Mobile phones
Answer: B

8. Internal testing assumes:

A. Public attacker
B. Insider access
C. No access
D. Only admin privileges
Answer: B

9. Remediation means:

A. Finding bugs
B. Fixing vulnerabilities
C. Creating attacks
D. Deleting logs
Answer: B

10. Final step of penetration testing:

A. Scanning
B. Reporting
C. Threat modeling
D. Logging
Answer: B

✅ UNIT–5: Hacking Techniques and Tools (10 MCQs)

1. Social engineering mainly exploits:

A. Hardware weaknesses
B. Human psychology
C. Network protocols
D. Encryption
Answer: B

2. SQL injection targets:

A. Database queries
B. CSS files
C. Tokens
D. Cookies only
Answer: A

3. XSS affects:

A. Server OS
B. Client browser
C. Network hardware
D. IoT devices
Answer: B

4. Broken authentication results in:

A. UI errors
B. Account compromise
C. Faster login
D. Better performance
Answer: B

5. CSRF forces users to:

A. Change passwords
B. Perform unwanted actions
C. Block sessions
D. Download files
Answer: B

6. Security misconfiguration includes:

A. Strong passwords
B. Default credentials
C. Disabled ports
D. Encrypted storage
Answer: B

7. Insecure cryptographic storage includes:

A. Proper hashing
B. Plain-text password storage
C. Using SSL
D. Tokenization
Answer: B

8. Injection occurs due to:

A. Secure code
B. Sanitized input
C. Poor input validation
D. Token usage
Answer: C

9. XSS allows attackers to:

A. Modify server hardware
B. Steal cookies
C. Install OS
D. Delete RAM
Answer: B

10. CSRF can be prevented using:

A. HTTP only cookies
B. CSRF tokens
C. DNS changes
D. USB firewall
Answer: B

✅ UNIT–6: Failure to Restrict URL Access & Tools (10 MCQs)

1. Failure to restrict URL access allows:

A. Faster browsing
B. Unauthorized access
C. Better UI
D. Faster caching
Answer: B

2. This tool is used for web app penetration testing:

A. MS Word
B. Burp Suite
C. Notepad++
D. Excel
Answer: B

3. Nikto is primarily used for:

A. Malware removal
B. Web server scanning
C. Antivirus scanning
D. Cloud storage
Answer: B

4. OpenVAS is a:

A. Code editor
B. Vulnerability scanner
C. Backup tool
D. Web server
Answer: B

5. Nexpose provides:

A. UI testing
B. Risk scoring
C. CSS validation
D. File editing
Answer: B

6. Burp Suite intercepts:

A. SMS
B. HTTP/HTTPS traffic
C. Bluetooth signals
D. Email logs
Answer: B

7. Comodo is known for:

A. SSL certificates
B. CSS templates
C. Database design
D. Mobile apps
Answer: A

8. URL access control must be checked:

A. Only at login
B. On every page request
C. Only for admin pages
D. Only for API calls
Answer: B

9. Direct URL access vulnerability is related to:

A. Authorization failure
B. Encryption failure
C. Cookie storage
D. DNS protocol
Answer: A

10. Proper access control uses:

A. Allow all roles
B. Role-based restrictions
C. Unlimited sessions
D. Plain HTTP
Answer: B