G H raisoni COLLEGE
Engineering and Management
Nagpur

raisoni EDUCATION

Web Application Security

History, Architecture, and Security Fundamentals

Introduction & History of Web Apps

- The Evolution:
- - Web 1.0 (Static): Read-only, simple HTML.
- - Web 2.0 (Dynamic): User-generated content, AJAX, social media.
- - Web 3.0 (Semantic): AI-driven, decentralized.

- The Shift: Moved from brochures to mission-critical business software.

Web Application Interface & Structure

- Interface:
- - Client-side (Browser) interaction via forms and scripts.

- N-Tier Architecture:
- - Presentation Layer (HTML/CSS/JS)
- - Logic Layer (Server-side: Node, Java, Python)
- - Data Layer (SQL/NoSQL Databases)

Benefits vs. Drawbacks

- Benefits:
- - Accessibility (Anywhere with internet)
- - Cross-Platform compatibility
- - Easier Maintenance (Server-side updates)

- Drawbacks:
- - Security Risks (Public exposure)
- - Performance (Internet dependent)

Web Application vs. Cloud Application

- Web Application:
- - Hosted on limited servers.
- - Focus on functionality delivered via browser.

- Cloud Application:
- - Scalability: Elastic scaling up/down.
- - Multitenancy: Single instance serves many users.
- - Resilience: High availability across regions.

Security Fundamentals: Validation & Surface

- Input Validation:
- - Rule: Never Trust User Input.
- - Syntactic vs. Semantic validation.
- - Use Whitelisting over Blacklisting.

- Attack Surface Reduction:
- - Minimize entry points.
- - Disable unused ports/features.
- - Remove default accounts.

Rules of Thumb & Threat Prioritization

- Rules of Thumb:
- - Defense in Depth (Layered security).
- - Least Privilege.
- - Fail Securely.

- Classifying Threats (STRIDE):
- - Spoofing, Tampering, Repudiation, Info Disclosure, DoS, Elevation of Privilege.

Origin Policy (SOP)

- Definition:
- - A script from one origin cannot access resources from another origin.

- Components of Origin:
- - Protocol (http/https), Domain, and Port.

- Purpose:
- - Prevents malicious sites from reading data from other open tabs (e.g., banking).

Exceptions to Same-Origin Policy

- 1. CORS (Cross-Origin Resource Sharing):
- - Servers use headers (Access-Control-Allow-Origin) to permit access.

- 2. JSONP (Historical):
- - Using <script> tags to fetch data (exempt from SOP).

- 3. HTML Tags:
- - Images, CSS, and Scripts can be loaded across domains.

Cross-Site Scripting (XSS)

- Definition:
- - Injecting malicious scripts into web pages viewed by others.

- Types:
- - Stored XSS (Database).
- - Reflected XSS (Immediate response).
- - DOM-based XSS (Client-side).

- Prevention: Sanitization and Content Security Policy (CSP).

Reflected XSS & HTML Injection

- Reflected XSS:
- - Malicious script is part of the request (e.g., URL parameters).
- - Server sends it back immediately.

- HTML Injection:
- - Injecting HTML tags (not necessarily scripts).
- - Used for defacement or fake login forms.

Cross-Site Request Forgery (CSRF)

- Concept:
- - Attacker tricks a logged-in user into executing unwanted actions.

- Example:
- - A hidden form submits a 'Transfer Money' request using the user's active cookies.

- Defense:
- - Anti-CSRF Tokens.
- - SameSite Cookie attribute.