**Module 10 Web Servers and Apps**

**10.0 Web Server**

**Web Server**

1. Server software/hardware that “serve” content to World Wide Web
2. Responds to client requests
3. Client requests for TCP connection (port 80 or 443)
4. Server waits for HTTP GET request from client to get content

**HTTP Request Methods**

1. GET
   1. Requests data from resource
   2. Can send data but data will be tagged in URL
2. HEAD
   1. Similar to GET
   2. Server must not return a message-body in response
   3. Method used for requesting headers/metadata
   4. Testing hypertext links for validity, accessibility & modifications
3. POST
   1. Request origin server accept the entity in request as a new subordinate of resource identified by the request-Line
   2. Function is determined by server
   3. Good method of submitting data to resource for processing
   4. Safer than GET when not stored in browser history
   5. Doesn’t display returned data in URL like GET
4. PUT
   1. Requests that stored entity stored under supplied Request URI
   2. Request-URI points to existing resource then enclosed entity shld be considered a modified version
5. DELETE – requests that origin server delete resource identified by request-URI
6. TRACE – used to invoke a remote, app layer loop back of request tunnel
7. CONNECT – reserved for use with a proxy that can dynamically switch to being a tunnel

**Web Server Attacks**

1. Directory Traversal
   1. Attempt to access restricted directories
   2. Sends HTTP requests asking server to drop back to root directory
   3. AKA as dot-dot-slash attack
   4. Eg. <http://www.test.com/../../../../etc/passwd>

**Web Server Attack Tools**

1. Metasploit
2. Website Mirroring – HTTrack

**OWASP Top 10 Web Application Security Risks**



**Injection Flaws**

1. Can result in data loss/corruption
2. Prevention
   1. Use safe API (Avoids Interpreter) – so code can’t run
   2. Whitelist server-side input validation
   3. Use SQL controls within queries to prevent mass disclosure of records in SQL Injection attacks

**Broken Authentication**

1. Can result in identity theft & fraud
2. Prevention
   1. Multifactor authentication
   2. No default credentials
   3. Check for weak passwords
      1. Follow NIST 800-63 Section 5.1.1 for password guidelines
   4. Harden against enumeration of accounts
   5. Limit failed login attempts

**Sensitive Data Exposure**

1. Can lead to identity theft
2. Prevention
   1. Classify data being process, stored & transmitted
   2. Apply appropriate controls
   3. Encrypt all data at rest & in transit

**XML External Entity (XXE)**

1. Weaknesses in XML processors
2. Can lead to data extraction, DoS, internal system scans
3. Prevention
   1. Use less complex data formats, like JSON
   2. Patch/upgrade all XML processors & libraries
   3. Disable XML external entity processing in XML documents & headers

**Broken Access Control**

1. Can cause admin privilege for attacker & users accessing/deleting records
2. Prevention
   1. Deny by default
   2. Disable web server directory listing
   3. Log access control failures

**Security Misconfiguration**

1. Can lead to unauthorised access/complete system compromise
2. Prevention
   1. Platform with no unnecessary features
   2. Hardening
   3. Segmented application architecture

**XSS (Cross-Site Scripting) Attacks**

1. Can lead to remote code execution on victim;s browser, stealing of credentials & delivery of malware to victim
2. Prevention
   1. Separating untrusted data from active browser content
   2. Escaping untrusted HTTP requesting data
   3. Enabling Content Security Policy (CSP)

**Insecure Deserialization**

1. Can lead to remote code execution
2. Prevention
   1. Implement integrity checks (Eg. digital signatures)
   2. Code isolation
   3. Logging deserialization exceptions & failures

**Using Components with Known Vulnerabilities**

1. Can lead to massive data breaches
2. Prevention
   1. Patching
   2. Only obtain from official sources
   3. Continuous inventory of client & server-side frameworks & libraries

**Insufficient Logging & Monitoring**

1. Can lead to successful exploits
2. Prevention
   1. Ensure all login & access control failures logged with sufficient context
   2. Ensure all logs are generated in easily consumed format
   3. Establish effective monitoring & alerts