2.0 Vulnerability Management

2.1 Given a scenario, implement vulnerability scanning methods and concepts.

- **Asset discovery**
  - Map scans
  - Device fingerprinting
- **Special considerations**
  - Scheduling
  - Operations
  - Performance
  - Sensitivity levels
  - Segmentation
  - Regulatory requirements
- **Internal vs. external scanning**
- **Agent vs. agentless**
- **Credentialed vs. non-credentialed**
- **Passive vs. active**
- **Static vs. dynamic**
  - Reverse engineering
  - Fuzzing
- **Critical infrastructure**
  - Operational technology (OT)
  - Industrial control systems (ICS)
  - Supervisory control and data acquisition (SCADA)
- **Security baseline scanning**
- **Industry frameworks**
  - Payment Card Industry Data Security Standard (PCI DSS)
  - Center for Internet Security (CIS) benchmarks
  - Open Web Application Security Project (OWASP)
  - International Organization for Standardization (ISO) 27000 series

2.2 Given a scenario, analyze output from vulnerability assessment tools.

- **Tools**
  - Network scanning and mapping
    - Angry IP Scanner
    - Maltego
  - Web application scanners
    - Burp Suite
    - Zed Attack Proxy (ZAP)
    - Arachni
    - Nikto
  - Vulnerability scanners
    - Nessus
    - OpenVAS
  - Debuggers
    - Immunity debugger
    - GNU debugger (GDB)
  - Multipurpose
    - Nmap
    - Metasploit framework (MSF)
    - Recon-ng
  - Cloud infrastructure assessment tools
    - Scout Suite
    - Prowler
    - Pacu

2.3 Given a scenario, analyze data to prioritize vulnerabilities.

- **Common Vulnerability Scoring System (CVSS) interpretation**
  - Attack vectors
  - Attack complexity
  - Privileges required
  - User interaction
  - Scope
  - Impact
    - Confidentiality
    - Integrity
    - Availability
- **Validation**
  - True/false positives
  - True/false negatives
- **Context awareness**
  - Internal
  - External
  - Isolated
- **Exploitability/weaponization**
- **Asset value**
- **Zero-day**

2.4 Given a scenario, recommend controls to mitigate attacks and software vulnerabilities.

- **Cross-site scripting**
  - Reflected
  - Persistent
- **Overflow vulnerabilities**
  - Buffer
  - Integer
  - Heap
  - Stack
- **Data poisoning**
- **Broken access control**
- **Cryptographic failures**
- **Injection flaws**
- **Cross-site request forgery**
- **Directory traversal**
- **Insecure design**
- **Security misconfiguration**
- **End-of-life or outdated components**
- **Identification and authentication failures**
- **Server-side request forgery**
- **Remote code execution**
- **Privilege escalation**
- **Local file inclusion (LFI)/remote file inclusion (RFI)**

2.5 Explain concepts related to vulnerability response, handling, and management.

- **Compensating control**
- **Control types**
  - Managerial
  - Operational
  - Technical
  - Preventative
  - Detective
  - Responsive
  - Corrective
- **Patching and configuration management**
  - Testing
  - Implementation
  - Rollback
  - Validation
- **Maintenance windows**
- **Exceptions**
- **Risk management principles**
  - Accept
  - Transfer
  - Avoid
  - Mitigate
- **Policies, governance, and service-level objectives (SLOs)**
- **Prioritization and escalation**
- **Attack surface management**
  - Edge discovery
  - Passive discovery
  - Security controls testing
  - Penetration testing and adversary emulation
  - Bug bounty
  - Attack surface reduction
- **Secure coding best practices**
  - Input validation
  - Output encoding
  - Session management
  - Authentication
  - Data protection
  - Parameterized queries
- **Secure software development life cycle (SDLC)**
- **Threat modeling**