Vulnerability Assessment and Penetration Testing

1. Vulnerability Scanning Techniques

1.1 Introduction to Vulnerability Scanning

Vulnerability scanning is the process of **identifying security weaknesses** in systems, networks, and applications using automated tools. It helps organizations understand their attack surface before an attacker exploits it. Vulnerability scanning is a **preventive security activity** and forms the foundation of vulnerability assessment and penetration testing.

1.2 Types of Vulnerability Scans

Network-Based Scanning
Network scans identify open ports, running services, and misconfigurations in networked systems. Tools like Nmap are commonly used to detect exposed services that could be exploited.

Application-Based Scanning
Application scans focus on web applications and identify vulnerabilities such as SQL injection, XSS, and insecure headers. Tools like Nikto and web scanners are used for this purpose.

Authenticated vs Unauthenticated Scanning

- **Unauthenticated scans** simulate an external attacker with no credentials.
- **Authenticated scans** use valid credentials to perform deeper analysis and identify internal vulnerabilities such as missing patches and weak configurations.

1.3 Vulnerability Scoring – CVSS

The Common Vulnerability Scoring System (CVSS) is used to **measure the severity** of vulnerabilities. CVSS provides a numerical score ranging from **0.0 to 10.0**, which helps in prioritizing remediation.

- Low: 0.1 – 3.9
- Medium: 4.0 – 6.9
- High: 7.0 – 8.9

- Critical: 9.0 – 10.0

For example, a Remote Code Execution vulnerability with a CVSS score of 8.8 is considered **High**, while critical vulnerabilities such as Apache Struts RCE may score above 9.0.

1.4 False Positives in Scanning

Automated scanners may sometimes report vulnerabilities that are **not actually exploitable**, known as false positives. Therefore, scan results must be validated manually through verification techniques such as service interaction and controlled testing.

1.5 Objective of Vulnerability Scanning

The primary objective of vulnerability scanning is to:

- Identify exposed services and weaknesses
- Assess risk using CVSS
- Provide accurate input for penetration testing
- Support informed remediation decisions

2. Penetration Testing Techniques

2.1 Introduction to Penetration Testing

Penetration testing is a **controlled and authorized security assessment** that simulates real-world attacks to evaluate the security posture of systems and applications. Unlike vulnerability scanning, penetration testing involves **active exploitation**.

2.2 Phases of Penetration Testing

Reconnaissance
Information gathering using OSINT techniques to identify domains, IP addresses, technologies, and exposed assets.

Scanning and Enumeration
Identifying open ports, services, and versions using scanning tools to expand the attack surface.

Exploitation
Actively exploiting identified vulnerabilities to gain unauthorized access.

Post-Exploitation
Maintaining access, privilege escalation, and collecting evidence.

Reporting
Documenting findings, risk impact, and remediation recommendations.

2.3 Penetration Testing Methodologies

Standard methodologies provide a structured approach to penetration testing. Common methodologies include:

- Penetration Testing Execution Standard (PTES)
- OWASP Web Security Testing Guide (WSTG)

These methodologies ensure consistent, ethical, and repeatable testing processes.

2.4 Ethics and Legal Considerations

Penetration testing must always be performed with **proper authorization** and within a defined scope. Unauthorized testing is illegal and unethical. Ethical hacking ensures confidentiality, integrity, and responsible disclosure of vulnerabilities.

3. Exploit Development Basics

3.1 Introduction to Exploits

An exploit is a piece of code or technique used to **take advantage of vulnerability**. Exploits demonstrate the real-world impact of security weaknesses.

3.2 Common Types of Exploits

Buffer Overflow
Occurs when a program writes more data than allocated memory, potentially allowing code execution.

SQL Injection
Occurs when untrusted input is improperly handled in database queries, allowing attackers to manipulate data.

Cross-Site Scripting (XSS)
Occurs when user input is not properly sanitized, allowing execution of malicious scripts in a victim's browser.

3.3 Proof of Concept (PoC) Exploits

Public exploit repositories provide Proof of Concept code to demonstrate vulnerabilities. These PoCs should be used **only in controlled lab environments** for learning and validation.

3.4 Security Mitigations

Modern systems implement mitigations such as:

- Address Space Layout Randomization (ASLR)
- Web Application Firewalls (WAF)
- Secure coding practices
- Regular patching and updates

These reduce the likelihood of successful exploitation.

4. Conclusion

Understanding vulnerability scanning, penetration testing methodologies, and exploit basics is essential for performing effective VAPT. Proper risk assessment, ethical conduct, and structured documentation are critical components of professional cybersecurity practice.