# **Example 1: Vulnerability Assessment of a Web Application**

## **1. Scope Definition**

• System: E-commerce web application

• Components: Web server, application server, database server

• Assessment Tools: Nmap, OWASP ZAP, Nessus

## **2. Data Collection**

• Network Scan: Using Nmap to identify open ports and services.

• Vulnerability Scan: Using Nessus to find known vulnerabilities.

• Web Application Scan: Using OWASP ZAP to test for web application vulnerabilities.

## **3. Performing the Assessment**

**Network Scan with Nmap**

nmap -sS -p 1-65535 -v -O -sV internetarchive.org

Results:

PORT STATE SERVICE VERSION

80/tcp open http Apache httpd 2.4.29

443/tcp open https Apache httpd 2.4.29

3306/tcp open mysql MySQL 5.7.21

**Web Application Scan with OWASP ZAP**

Results:

• SQL Injection: Found in the login form, parameter username

• Cross-Site Scripting (XSS): Found in the search functionality

• Cross-Site Request Forgery (CSRF): Missing CSRF tokens on critical actions

## **4. Data Analysis**

• **SQL Injection:**

o Impact: Unauthorized data access, data modification, potential full system

compromise

o Fix: Use prepared statements and parameterized queries

• **Cross-Site Scripting (XSS):**

o Impact: User session hijacking, defacement, and phishing

o Fix: Implement input validation and output encoding

• **Cross-Site Request Forgery (CSRF):**

o Impact: Unauthorized actions performed on behalf of authenticated users

o Fix: Implement CSRF tokens in forms and critical actions

## **5. Reporting**

Executive Summary: The e-commerce web application is exposed to several high-risk

vulnerabilities, including SQL injection, and XSS. Immediate actions

should be taken to patch the systems and secure the application to prevent potential exploits.

**Detailed Report:**

• **Vulnerabilities Found:**

o SQL Injection in login form

o Cross-Site Scripting in search functionality

o Cross-Site Request Forgery in critical actions

• **Recommendations:**

o Implement prepared statements for database interactions

o Validate and sanitize user inputs

o Implement CSRF protection mechanisms

## **6. Remediation**

• Code Fixes:

o Implement secure coding practices for SQL queries and input handling

• Configuration Changes:

o Harden server configurations to reduce attack surface

• Monitoring:

o Set up continuous monitoring and regular vulnerability scans to detect and

address new vulnerabilities promptly