INT310: Web Application Source Code Vulnerability Analysis – Lab 2: Comprehensive Vulnerability Review

Objective

The goal of this lab is to simulate real-world vulnerability assessments by reviewing PHP and Python scripts for security issues. You will identify vulnerabilities using the **Common Weakness Enumeration (CWE)** list and patch the code to improve its security. This exercise provides hands-on experience that strengthens your secure coding skills—an essential capability in the cybersecurity industry.

Instructions

1. **Review Code**: Analyze the provided PHP and Python code files for security vulnerabilities.
   - **PHP File**: vulnerable_script.php
   - **Python File**: vulnerable_script.py
2. **Identify Security Vulnerabilities**: Use the CWE list to help you identify common vulnerabilities in the code.
   - Examine the logic, input handling, and the overall structure for potential weak spots.
   - **Document Findings**: Fill out the provided worksheet with detailed information on each identified vulnerability:
     - CWE Number
     - File Name
     - Line Number
     - Description of the Vulnerability (how it could be exploited and its potential impact).
3. **Patch the Vulnerabilities**: Apply secure coding practices to patch all identified issues.
   - Ensure your fixes follow industry best practices and are consistent with modern security standards.
   - Patch the code **without altering the intended functionality**.

4. **Submit the Corrected Code and Worksheet**: Zip your completed assignment into a single file, ensuring it contains:
    - The corrected PHP and Python files with vulnerabilities fixed.
    - The fully completed worksheet documenting all identified vulnerabilities.

Lab Deliverables

1. **Completed Worksheet**:
    - Document each identified vulnerability clearly.
    - Example:

CWE	File Name	Line #	Description of Vulnerability
89	vulnerable_script.php	4	SQL Injection vulnerability due to unparameterized query.
78	vulnerable_script.php	9	OS Command Injection via unsanitized user input.
79	vulnerable_script.py	13	Cross-site scripting (XSS) risk from improper input validation.

2. **Patched PHP and Python Scripts**: Ensure that:
    - All identified vulnerabilities are properly patched.
    - Code comments are added to highlight the changes and improvements made.
3. **ZIP File Submission**:
    - ZIP file must contain the **corrected PHP and Python code files** and the **completed worksheet**.

CWE Categories to Consider

Ensure that you identify at least 20 vulnerabilities from the following list:

- **CWE-89**: SQL Injection
- **CWE-78**: OS Command Injection

- **CWE-120**: Buffer Overflow
- **CWE-79**: Cross-site Scripting (XSS)
- **CWE-306**: Missing Authentication for Critical Function
- **CWE-862**: Missing Authorization
- **CWE-798**: Use of Hard-coded Credentials
- **CWE-311**: Missing Encryption of Sensitive Data
- **CWE-434**: Unrestricted File Upload
- **CWE-807**: Reliance on Untrusted Inputs in a Security Decision
- **CWE-250**: Execution with Unnecessary Privileges
- **CWE-352**: Cross-Site Request Forgery (CSRF)
- **CWE-22**: Path Traversal
- **CWE-494**: Download of Code Without Integrity Check
- **CWE-863**: Incorrect Authorization
- **CWE-829**: Inclusion of Functionality from Untrusted Control Sphere
- **CWE-732**: Incorrect Permission Assignment for Critical Resource
- **CWE-676**: Use of Potentially Dangerous Function
- **CWE-327**: Use of a Broken or Risky Cryptographic Algorithm
- **CWE-131**: Incorrect Calculation of Buffer Size
- **CWE-307**: Improper Restriction of Excessive Authentication Attempts
- **CWE-601**: URL Redirection to Untrusted Site
- **CWE-134**: Uncontrolled Format String
- **CWE-190**: Integer Overflow or Wraparound
- **CWE-759**: Use of a One-Way Hash Without a Salt

Grading Criteria

This assignment will be reviewed by external cybersecurity professionals, so ensure your submission meets the highest standards. Graders will focus on:

1. **Accuracy of Identified Vulnerabilities**: Did you correctly identify the most critical issues in the code?
2. **Quality of Patches**: Are your fixes secure and in line with industry best practices?
3. **Clarity of Documentation**: Is your worksheet clear and detailed, with correct CWE classifications?
4. **Professionalism**: Demonstrating professional integrity by completing the work without relying on AI code generators.

Warnings

- **AI Usage**: Remember that using AI tools like ChatGPT or code generators for this assignment will hinder your learning. Industry professionals will review your work, so submitting AI-generated responses could damage your reputation.
- **External Graders**: Your work will be evaluated by experienced cybersecurity professionals, not just your course tutor. Take this assignment seriously—this is an opportunity to showcase your skills to potential mentors or future employers.

Final Note

This assignment is designed to provide you with hands-on experience in identifying and patching security vulnerabilities. By completing it, you will strengthen your secure coding practices, a critical competency in cybersecurity.

Your professional reputation in the field depends on these skills, and this lab is an excellent opportunity to demonstrate your abilities. Be thorough, take pride in your work, and treat this as if your career depends on it—because one day, it might.

Good Luck!