Homework 3

Demonstrating Risky Resource Management

Overview

In this homework you use AWS Cloud 9 to create two different unique and complete demonstrations of Risky Resource Management such as Buffer Copy without Checking Size of Input, Improper Limitation of a Pathname to a Restricted Directory, Download of Code Without Integrity Check, Inclusion of Functionality from Untrusted Control Sphere, Use of Potentially Dangerous Function, Incorrect Calculation of Buffer Size, Uncontrolled Format String, and Integer Overflow or Wraparound.

You will provide unique code that contains the vulnerability and then provide an updated version of the code that fixes the vulnerability. You should also describe why the original code was vulnerable and discuss specific attack methods a user could try to exploit the vulnerability. Finally discuss how the new code fixes the vulnerability.

Assignment Details

Be sure you have carefully read and understand the materials in weeks 3 and 4.

1. Select 2 CWE/SANS Top 25 vulnerabilities under the category of **Risky Resource Management** from one of these specific issues**:**
    a. Buffer Copy without Checking Size of Input,
    b. Improper Limitation of a Pathname to a Restricted Directory ,
    c. Download of Code Without Integrity Check, and
    d. Inclusion of Functionality from Untrusted Control Sphere**.**
    e. Use of Potentially Dangerous Function
    f. Incorrect Calculation of Buffer Size
    g. Uncontrolled Format String
    h. Integer Overflow or Wraparound

   Review and try the existing examples in links in the classroom. Use AWS Cloud 9 to experiment. Work in multiple languages where possible.

2. Using AWS Cloud 9, create your own **unique** example for each of the 2 vulnerabilities in this category.
3. Your code examples do not need to large or fully functional from an application standpoint. However; they need to include all of the code such that the vulnerability can be fully explained and corrected.
4. Use the information in the CWE/SANS Top 25 vulnerabilities to understand and experiment.
5. Be sure your documentation and descriptions are detailed and completed.
6. You may need to conduct additional research to better understand the vulnerability or the features associated with a specific language.

Deliverables

Provide all of your source files for this assignment along with your well-organized documentation (word of PDF file) supporting the files. Be sure to provide all documentation in one word or PDF document. You can compress the source files and documentation into a zip application for easier upload.

Grading Rubric:

Attribute	Meets
Vulnerabilities	**50 points** Selects 2 CWE/SANS Top 25 vulnerabilities under the category of Risky Resource Management. (10 points) Creates **unique** example for each of the 2 vulnerabilities in this category. (20 points) Demonstrates for each application, they are vulnerable to an attack. (20 points)
Mitigation	**25 points** Fixes the issues in each of the two examples you created. (25 points)
Documentation and submissions	**25 points** Provides all source files (those with vulnerabilities, those fixed and any supplemental files needed to run the application. (15 points) Within a word or PDF file, documents vulnerabilities and describe specifically how the issues were corrected. (10 points)