**Homework – Week 2**

Student’s Name or Student’s Names

Department Affiliation, University Affiliation

Course Number: Course Name

Instructor’s Name

Assignment Due Date

**Homework – Week 2**

1. **Dangerous Software Weaknesses**
2. **CWE-787 Out-of-bounds Write**

Based on this weakness, the software writes data past the end or before the beginning of the intended buffer, which may amount into corruption of the data, crash or code execution. Additionally, this may amount to writer operation that is producing unexpected results. Ideally, in programming defensively to neutralize this attack, one of the mechanisms that can be adopted is environment hardening by running and compiling software through features or extensions that randomly arrange the program position’s executable and libraries in memory. Another way of preventing this form of weakness on the implementation stage through replacement of unbound copy functions with analogous functions that support length arguments. This will in turn prevent buffer overflow attacks.

1. **CWE-78 Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection')**

Ideally, this is a software weakness which encompasses software constructing all or parts of operating system command using externally or internally influenced inputs from upstream components, but nor possible to neutralize or incorrectly neutralizes special elements that can do modification of the intended OS command when sent to a downstream component. From the above weakness standpoint, these variants capture distinct programming errors. In preventing these weaknesses, the idea is to ensure proper neutralization of special elements used in a command. This would prevent command injection as well as argument injection. Another prevention measure of this weakness is by ensuring that programmers have adopted proper architectural security design.

1. **CWE-22 Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal')**

Based on this kind of weakness, the software uses external inputs in constructing the pathname that is intended to identify a file or directory, which is based on underneath a restricted parent directory, but the software is not able to neutralize special elements in the pathname in resolving to a location that is outside the restricted area. This weakness can be resolved through various ways such as use of correctly resolved names or reference, proper input validation of the program, proper testing of the coding errors. Lastly, this weakness can be resolved by ensuring that the code written cannot be rewritten again by the attacker making all the application directories invisible.

1. **CWE-807 Reliance on Untrusted Inputs in a Security Decision**

Notably, this form of software weakness is common, and it entails application use of protection mechanism that relies on relying on the existence values of an input; however, the input can be modified by untrusted actor in a manner that bypasses the protection mechanism. Attackers through this form of weakness can make changes of the inputs via customized clients or other form of attacks. This software weakness can be prevented by ensuring there is no authentication bypass and ensuring proper validation and integrity checks in security decision. Also, developers should ensure there is use of proper architectural design during the development.

1. **The code below is affected by a weakness. What is the weakness? Choose the best answer**



Answer: (D) CWE-787 – Out-of-bounds Write

1. The code below is affected by a weakness. What is the weakness? Choose the best answer



Answer: (B) CWE-807 - Reliance on Untrusted Inputs in a Security Decision

1. (1 Pts.) The code below is affected by a weakness. What is the weakness? Choose the best answer.



Answer: } (D) CWE-787 – Out-of-bounds Write

1. . (1 Pts.) The code below is affected by a weakness. What is the weakness? Choose the best answer.



**Answer:** (A) CWE-78 - OS Command Injection

**References**

Common Weakness Enumeration (n.d). A Community-Developed List of Software and Hardware Weakness Types