**Chapter 11 – Software Security**

**TRUE/FALSE QUESTIONS:**

T F 1. Many computer security vulnerabilities result from poor programming

practices.

T F 2. Security flaws occur as a consequence of sufficient checking and

validation of data and error codes in programs.

T F 3. Software security is closely related to software quality and reliability.

T F 4. A difference between defensive programming and normal practices is

that everything is assumed.

T F 5. Programmers often make assumptions about the type of inputs a

program will receive.

T F 6. Defensive programming requires a changed mindset to traditional

programming practices.

T F 7. To counter XSS attacks a defensive programmer needs to explicitly

identify any assumptions as to the form of input and to verify that any

input data conform to those assumptions before any use of the data.

T F 8. Injection attacks variants can occur whenever one program invokes the

services of another program, service, or function and passes to it

externally sourced, potentially untrusted information without sufficient

inspection and validation of it.

T F 9. Cross-site scripting attacks attempt to bypass the browser’s security

checks to gain elevated access privileges to sensitive data belonging to

another site.

T F 10. To prevent XSS attacks any user supplied input should be examined

and any dangerous code removed or escaped to block its execution.

T F 11. An ASCII character can be encoded as a 1 to 4 byte sequence using

the UTF-8 encoding.

T F 12. There is a problem anticipating and testing for all potential types of

non-standard inputs that might be exploited by an attacker to subvert a program.

T F 13. Key issues from a software security perspective are whether the

implemented algorithm correctly solves the specified problem, whether the machine instructions executed correctly represent the high level algorithm specification, and whether the manipulation of data values in variables is valid and meaningful.

T F 14. Without suitable synchronization of accesses it is possible that values

may be corrupted, or changes lost, due to over-lapping access, use, and replacement of shared values.

T F 15. The correct implementation in the case of an atomic operation is to

test separately for the presence of the lockfile and to not always attempt to create it.

**MULTIPLE CHOICE QUESTIONS:**

1. “Incorrect Calculation of Buffer Size” is in the \_\_\_\_\_\_\_\_\_\_ software error category.

A. Porous Defenses

B. Allocation of Resources

C. Risky Resource Management

D. Insecure Interaction Between Components

1. “Improper Access Control (Authorization)” is in the \_\_\_\_\_\_\_\_\_ software error

category.

A. Porous Defenses

B. Allocation of Resources

C. Risky Resource Management

D. Insecure Interaction Between Components

1. Defensive programming is sometimes referred to as \_\_\_\_\_\_\_\_\_.

A. variable programming B. secure programming

C. interpretive programming D. chroot programming

1. Incorrect handling of program \_\_\_\_\_\_\_ is one of the most common failings in

software security.

A. lines B. input

C. output D. disciplines

1. \_\_\_\_\_\_\_\_\_ is a program flaw that occurs when program input data can accidentally or deliberately influence the flow of execution of the program.

A. PHP attack B. Format string injection attack

C. XSSattack D. Injection attack

1. A \_\_\_\_\_\_\_\_\_ attack occurs when the input is used in the construction of a command that is subsequently executed by the system with the privileges of the Web server.

A. command injection B. SQL injection

C. code injection D. PHP remote code injection

1. A \_\_\_\_\_\_\_ attack is where the input includes code that is then executed by the attacked system.

A. SQL injection B. cross-site scripting

C. code injection D. interpreter injection

1. Blocking assignment of form field values to global variables is one of the defenses available to prevent a \_\_\_\_\_\_\_\_\_\_ attack.

A. PHP remote code injection B. mail injection

C. command injection D. SQL injection

1. \_\_\_\_\_\_\_\_\_\_ attacks are vulnerabilities involving the inclusion of script code in the HTML content of a Web page displayed by a user’s browser.

A. PHP file inclusion B. Mail injection

C. Code injection D. Cross-site scripting

1. A \_\_\_\_\_\_\_\_ is a pattern composed of a sequence of characters that describe allowable input variants.

A. canonicalization B. race condition

C. regular expression D. shell script

1. The intent of \_\_\_\_\_\_\_\_ is to determine whether the program or function correctly handles all abnormal inputs or whether it crashes or otherwise fails to respond appropriately.

A. shell scripting B. fuzzing

C. canonicalization D. deadlocking

1. A stead reduction in memory available on the heap to the point where it is completely exhausted is known as a \_\_\_\_\_\_\_\_.

A. fuzzing B. deadlock

C. memory injection D. memory leak

1. The most common technique for using an appropriate synchronization mechanism to serialize the accesses to prevent errors is to acquire a \_\_\_\_\_\_\_ on the shared file, ensuring that each process has appropriate access in turn.

A. lock B. code injection

C. chroot jail D. privilege escalation

14. \_\_\_\_\_\_\_\_\_ are a collection of string values inherited by each process from its parent that can affect the way a running process behaves.

A. Deadlocks B. Privileges

C. Environment variables D. Race conditions

15. The most common variant of injecting malicious script content into pages returned to users by the targeted sites is the \_\_\_\_\_\_\_\_\_ vulnerability.

A. XSS reflection B. chroot jail

C. atomic bomb D. PHP file inclusion

**SHORT ANSWER QUESTIONS:**

1. “Failure to Preserve SQL Query Structure” is in the \_\_\_\_\_\_\_\_\_\_ CWE/SANS software error category.
2. \_\_\_\_\_\_\_\_\_\_ programming is a form of design intended to ensure the continuing function of a piece of software despite unforeseeable usage of the software.
3. Program \_\_\_\_\_\_\_ refers to any source of data that originates outside the program and whose value is not explicitly known by the programmer when the code was written.
4. Two key areas of concern for any input are the \_\_\_\_\_\_\_ of the input and the meaning and interpretation of the input.
5. A number of widely used standard C \_\_\_\_\_\_\_\_\_ compound the problem of buffer overflow by not providing any means of limiting the amount of data transferred to the space available in the buffer.
6. Program input data may be broadly classified as textual or \_\_\_\_\_\_.
7. In the \_\_\_\_\_\_\_\_ attack the user supplied input is used to construct a SQL request to retrieve information from a database.
8. \_\_\_\_\_\_\_\_\_ attacks are most commonly seen in scripted Web applications.
9. A variant where the attacker includes malicious script content in data supplied to a site is the \_\_\_\_\_\_\_\_\_\_ vulnerability.
10. The process of transforming input data that involves replacing alternate, equivalent encodings by one common value is called \_\_\_\_\_\_\_\_\_.
11. The major advantage of \_\_\_\_\_\_\_\_ is its simplicity and its freedom from assumptions about the expected input to any program, service, or function.
12. A \_\_\_\_\_\_\_\_ occurs when multiple processes and threads compete to gain uncontrolled access to some resource.
13. UNIX related systems provide the chroot system function to limit a program’s view of the file system to just one carefully configured section that is known as a \_\_\_\_\_\_\_\_.
14. If privileges are greater than those already available to the attacker the result is a \_\_\_\_\_\_\_\_\_.
15. The principle of \_\_\_\_\_\_\_\_ strongly suggests that programs should execute with the least amount of privileges needed to complete their function.