Research
Week 11

CIT 288

Chaz Davis

BCTC

Spring 2020

March 7, 2020

Chaz Davis March 7, 2020

Chapter 11

i) Define defensive programming.

Defensive programming is when a programmer anticipates problems and writes code to deal with them. That said, the whole point of defensive programming is guarding against errors you don't expect. A defensive programmer is on the watch for trouble, to avoid it before it can cause real problems. The idea is not to write code that never fails. That is a utopian dream. The idea to make the code fail beautifully in case of any unexpected issue. Fail beautifully can mean any one of the following.

- Fail Early
- Fail Safe
- Fail Clearly

ii) Define input fuzzing. State where this technique should be used.

FUZZ TESTING (fuzzing) is a software testing technique that inputs invalid or random data called FUZZ into the software system to discover coding errors and security loopholes. Data is inputted using automated or semi-automated testing techniques after which the system is monitored for various exceptions, such as crashing down of the system or failing built-in code, etc.

Fuzz testing was originally developed by Barton Miller at the University of Wisconsin in 1989. Fuzz testing or fuzzing is a Software testing technique, and it is a type of Security Testing.

iii) List several software security concerns associated writing safe program code.

- Correct Algorithm Implementation
- Ensuring that machine language corresponds to the algorithm
- Correct interpretation of data values
- Correct use of memory
- Preventing race conditions with shared memory

iv) Define the principle of least privilege.

Programs should execute with the least amount of privilges needed to complete their function.

v) Identify several issues associated with the correct creation and use of a temporary file in a shared directory.

The temporary file must not be accessed by another process. An attacker could guess the name of the temporary file and create it in between the time the program checks if it exists and subsequently creating it. The program could be redirected and would overwrite an existing file. Thus the use of secure system calls is advised to avoid race conditions.