

Lecture 27: December 2, 2019

Reading: *text*, §24.4.2–24.5

Assignments: Homework 5, due on December 6, 2019 at 11:59pm
Lab 3, due on December 6, 2019 at 11:59pm

1. Greetings and felicitations!
2. PA Model (Neumann's organization)
 - (a) Goal: develop techniques to search for vulnerabilities that less experienced people could use
 - (b) Improper protection (initialization and enforcement)
 - i. Improper choice of initial protection domain: incorrect initial assignment of security or integrity level at system initialization or generation; a security critical function manipulating critical data directly accessible to the user;
 - ii. Improper isolation of implementation detail: allowing users to bypass operating system controls and write to absolute input/output addresses; direct manipulation of a hidden data structure such as a directory file being written to as if it were a regular file; drawing inferences from paging activity
 - iii. Improper change: the time-of-check to time-of-use flaw; changing a parameter unexpectedly;
 - iv. Improper naming: allowing two different objects to have the same name, resulting in confusion over which is referenced;
 - v. Improper deallocation or deletion: leaving old data in memory deallocated by one process and reallocated to another process, enabling the second process to access the information used by the first; failing to end a session properly
 - (c) Improper validation: not checking critical conditions and parameters, so a process addresses memory not in its memory space by referencing through an out-of-bounds pointer value; allowing type clashes; overflows
 - (d) Improper synchronization
 - i. Improper indivisibility: interrupting atomic operations (e.g. locking); cache inconsistency
 - ii. Improper sequencing: allowing actions in an incorrect order (e.g. reading during writing)
 - (e) Improper choice of operand or operation: using unfair scheduling algorithms that block certain processes or users from running; using the wrong function or wrong arguments.
3. NRL
 - (a) Goal: Find out how vulnerabilities enter the system, when they enter the system, and where they are
 - (b) Axis 1: inadvertent (RISOS classes) vs. intentional (malicious/nonmalicious)
 - (c) Axis 2: time of introduction (development, maintenance, operation)
 - (d) Axis 3: location (hardware, software: OS, support utilities, applications)Aslam
 - (a) Goal: Treat vulnerabilities as faults
 - (b) Coding faults: introduced during software development
 - i. Synchronization errors
 - ii. Validation errors
 - (c) Emergent faults: introduced by incorrect initialization, use, or application
 - i. Configuration errors
 - ii. Environment faults
 - (d) Introduced decision procedure to classify vulnerabilities in exactly one category
4. Some common vulnerabilities
 - (a) Catalogues: CVE (Common Vulnerabilities and Exposures), CWE (Common Weakness Enumeration)

- (b) 2011 MITRE/SANS Top 25 Most Dangerous Software Errors
- (c) OWASP Top 10 – 2017 The Ten Most Critical Web Application Security Risks