Lecture 9 Outline

Reading: *text*, §23.4 **Assignments due:** Homework #2, due April 27, 2011 at 11:55pm

Discussion Problem. How does weapon development compare to developing computer security mechanisms?

> Weapons developers, when given a choice, always go for the complex, elaborate solution at the expense of the simple one. Complexity leads to higher costs: purchase costs, operations costs, and maintenance costs. Higher costs result in fewer weapons, which, in turn, lead to contrived tests and analyses to prove that the relatively few complex systems can overcome the larger numbers of the simpler, less expensive weapons of the enemy. The fewer the weapons, the tighter is the control of these precious assets by a centralized command structure. The elaborate paraphernalia that comes with the centralized command structure only adds to the complexity of the overall system.[1]

1. 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; 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; 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.
2. 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)

[1] J. Burton, *The Pentagon Wars*, Naval Institute Press, Annapolis, MD (1993), p. 41.