# Lecture 7, April 15

**Reading:** §23.1–4, [PTE12][1] **Assignments due:** Homework #1, *late; 20% off*: April 15, 2013 at 11:55pm
Homework #2, due April 26, 2013 at 11:55pm

***Discussion question***. How does weapon development, as described in the following paragraph, 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.[2]

***Lecture outline***.

1. Where to start
   a. Unknown system
   b. Known system, no authorized access
   c. Known system, authorized access
2. Examples
   a. Burroughs system
   b. Corporate site
3. Vulnerability models
   a. PA model
   b. RISOS
   c. NRL
   d. Aslam
4. Example Flaws
   a. *fingerd* buffer overflow
   b. *xterm* race condition
5. RISOS
   a. Goal: Aid managers, others in understanding security issues in OSes, and work required to make them more secure
   b. Incomplete parameter validation—failing to check that a parameter used as an array index is in the range of the array;
   c. Inconsistent parameter validation—if a routine allowing shared access to files accepts blanks in a file name, but no other file manipulation routine (such as a routine to revoke shared access) will accept them;
   d. Implicit sharing of privileged/confidential data—sending information by modulating the load average of the system;
   e. Asynchronous validation/Inadequate serialization—checking a file for access permission and opening it non-atomically, thereby allowing another process to change the binding of the name to the data between the check and the open;
   f. Inadequate identification/authentication/authorization—running a system program identified only by name, and having a different program with the same name executed;
   g. Violable prohibition/limit—being able to manipulate data outside one's protection domain; and
   h. Exploitable logic error—preventing a program from opening a critical file, causing the program to execute an error routine that gives the user unauthorized rights.

[1] These are available in the Resources area of SmartSite; look in the folder "Handouts"
[2] J. Burton, *The Pentagon Wars*, Naval Institute Press, Annapolis, MD (1993), p. 41.