

Lecture 8, April 17

Reading: §23.1–4

Assignments due: Homework #2, due April 26, 2013 at 11:55pm

Discussion question. What does the following say to an attacker trying to break into a computer system? To the system administrator or security officer trying to defend that system?

All warfare is based on deception. Hence, when able to attack, we must seem unable; when using our forces, we must seem inactive; when we are near, we must make the enemy believe we are far away; when far away, we must make him believe we are near. Hold out baits to entice the enemy. Feign disorder, and crush him. If he is secure at all points, be prepared for him. If he is in superior strength, evade him. If your opponent is of a choleric temper, seek to irritate him. Pretend to be weak, that he may grow arrogant. If he is taking his ease, give him no rest. If his forces are united, separate them. Attack him where he is unprepared, appear where you are not expected.¹

Lecture outline.

1. 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.

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; 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)

¹Sun Tzu, *The Art of War*, James Clavell, ed., Dell Publishing, New York, NY (1983), p. 11.

- 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)
4. 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