Lecture 4, April 8

Reading: [Chr11,OWA13][1] **Assignments due:** Homework #1, due April 12, 2013

Discussion question. Microsoft spent February of 2003 teaching its programmers how to check their code for security vulnerabilities and how to introduce common security flaws. Yet many Microsoft programs still have security vulnerabilities. What problems do you think Microsoft encountered, and will encounter, in trying to find and clean up the vulnerabilities in its systems?

Lecture outline.

1. Greetings and felicitations!
   a. If you have not completed the prerequisites *in an earlier term*, please fill out the wavier form and send it to me — otherwise the department ***will*** drop you from the course!
2. 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 – 2013 rc1 The Ten Most Critical Web Application Security Risks
3. MITRE/SANS list
   a. Insecure interactions among components
      i. SQL injection
      ii. OS command injection
      iii. Cross-site scripting
      iv. Unrestricted upload of file with dangerous type
      v. Cross-site request forgery
      vi. URL redirect to untrusted site
   b. Risky resource management
      i. Buffer copy without checking size of input
      ii. Improper limitation of a pathname to a restricted directory
      iii. Download of code without integrity check
      iv. Inclusion of functionality from untrusted control sphere
      v. Use of potentially dangerous function
      vi. Incorrect calculation of buffer size
      vii. Uncontrolled format string
      viii. Integer overflow or wraparound
   c. Porous defenses
      i. Missing authentication for critical function
      ii. Missing authorization
      iii. Use of hard-coded credentials
      iv. Missing encryption of sensitive data
      v. Reliance on untrusted inputs in a security decision
      vi. Execution with unnecessary privileges
      vii. Incorrect authorization
      viii. Incorrect permission assignment for critical resource
      ix. Use of a broken or risky cryptographic algorithm
      x. Improper restriction of excessive authentication attempts
      xi. Use of a one-way hash without a salt
4. OWASP list
   a. Injection
   b. Broken authentication and session management
   c. Cross-site scripting
   d. Insecure direct object references

[1] These are available in the Resources area of SmartSite; look in the folder "Handouts"

   e. Security misconfiguration
   f. Sensitive data exposure
   g. Missing function level access control
   h. Cross-site request forgery
   i. Using known vulnerable components
   j. Unvalidated redirects and forwards
5. Comparison
   a. Everything on the OWASP list is also on the MITRE/SANS list
   b. Injection is #1 on both lists
   c. The MITRE/SANS list covers vulnerabilities generally; OWASP covers only web vulnerabilities