

CS 4330-01 Building Secure Software
Winter 2008
MWF 9:20–10:30 a.m., Sc S 125

Instructor: William R. Nico

Office: Robinson 239

Phone: (510)885-3386

E-mail: nico@csueastbay.edu (On the mcs system just “nico” will work.)

Web: www.mcs.csueastbay.edu/~nico

Office hours: MW 1:00–2:30 p.m. or by appointment.

Text 1: J. Viega and G. McGraw, *Building Secure Software*, Addison-Wesley, 2002.

Text 2: G McGraw, *Software Security: Building Security In*, Addison-Wesley, 2006.

Recommended: G. Hoglund and G. McGraw, *Exploiting Software: How to Break Code*, Addison-Wesley, 2004.

Note: These three volumes are available as a boxed set at a discounted price.

Software flaws—either design flaws or implementation flaws—are the root cause of the security problems so widely observed in recent years. One reason is that most development—design, implementation, and testing—has been concerned primarily with *features* and has not directly addressed *security* issues.

The purpose of this course is to investigate security issues in software design and implementation. It will explore both high-level and low-level issues that need to be addressed. This means that it will range over a large area of computer science. The more knowledge one brings to such a course, the more valuable it can be. However, a good understanding of software, say, at the *Data Structures (CS 3240)* level, should be adequate preparation for this course.

Activities in the course will include readings in the two texts and perhaps other sources and will involve programs experimenting with the concepts encountered in the course.

Grading: The course grade will be computed roughly as follows. (The date of the midterm is subject to change. Any change will be announced in class.)

- Written homework and programs (As assigned) 15%
- Midterm (Friday, February 8) 35%
- Final exam (Monday, March 17, 9:00–11:50 a.m.) 50%

Late homework and programs will *not* be accepted. Homework is to be turned in at the *beginning* of class on the due date unless the assignment states otherwise.

Written work: Any handwritten work submitted for the course, *including in-class tests*, must be done in *ink!*

Advice and Consultation: The programming projects are to be *individual* efforts, not group efforts. This means that there should be no sharing of code; such sharing constitutes *academic dishonesty*, as described in the *CSUEB Catalog*. “High level” discussion of algorithms (such as takes place in class) is acceptable, but detailed discussion is not. Any essential code included from sample programs must be properly acknowledged in comments. Any work not your own, e.g., results obtained from reference sources or other individuals, should receive appropriate bibliographic *citations*. *Plagiarism* will but subject to appropriate penalties, up to and including failure for the course.

Test policy: Students should be prepared to display a current photo id at test times, if asked.

Make-up policy: Make-up tests will be considered only in *unusual* circumstances, and then only if arrangements have been made in *advance*.