

Two Cool New Software Assurance Initiatives – Need Your Help

- Getting 100 Colleges to Teach Secure Coding in Core Courses in 2009
- Measuring The Impact Of Improved Software Development Processes



The Most Trusted Name in
Information Security



Teaching Secure Coding In The Core Curriculum in Colleges

- Teaching secure coding is one of the few “proactive” security initiatives.
- Few students write robust programs
 - Curriculum already crowded
 - Emphasis in most courses on getting programs working right
 - Tenured faculty apparently WILL NOT teach secure coding in core courses – even when they are given money to build curriculum.
- How can we improve quality of programs that students write throughout undergraduate, graduate work?
 - In particular, how can we get students to think about security considerations?

How to Do It, Approach 1

- Add security to exercises for general classes
 - Intro programming: integer or buffer overflow
 - Database: something on SQL injection
 - Programming languages: type clashes
 - Operating systems: race conditions
- Workshop held in April looked at ways to do this
 - Web site under development
 - Proposal for future workshop being developed

How to Do It, Approach 2

- Students must know how to write
 - Critical in all majors requiring communication, literary analysis skills
- Many don't
 - Majors provide support for writing in classes (law, English, rhetoric, *etc.*)
- Does not add material to curriculum
 - Instructors focus on content, not mechanics
 - Provides reinforcement

Secure Programming Clinic

- Genesis: operating system class
 - TA deducted for poor programming style
 - Dramatic improvement in quality of code!
- Programming foundational in CS
 - Just like writing is in English (and, really, all majors ...)
 - Clinicians assume students know some elements of style
 - Level of students affect what clinic teaches

How the Clinic Functions

- Assist students
 - Clinicians examine program, meet with student to give feedback
 - Clinic does not grade style
- Assist instructors
 - Clinic grades programs' styles
 - Meet with students to explain grade, how the program should have been done
 - Class readers can focus on program *correctness* (as defined by assignment)

Some Experience

- Tested in computer security class
 - Class emphasizes robust, secure programming
- Setup for class
 - Class had to analyze small program for security problems
 - Class applied Fortify code analysis tool to larger program, and traced attack paths
 - Thanks to Fortify for providing access to the tool!

How It Worked

- Write program to check attributes of file; if correct, change ownership, permissions
 - If done wrong, leads to TOCTTOU flaw
- Students had to get program checked at clinic before submitting it
 - Students sent program to clinician first
 - Clinician reviewed program before meeting with student
 - Student then could modify program

Results

Programming Problem	Before	After
TOCTTOU race condition	100%	12%
Unsafe calls (<i>strcpy</i> , <i>strcat</i> , etc.)	53%	12%
Format string vulnerability	18%	0%
Unnecessary code	59%	53%
Failure to zero out password	70%	0%
No sanity checking on modification time	82%	35%
Poor style	41%	N/A

100 Schools In 2009: The Two Ways You Can Help

- Provide funding to a school that trains programmers you hire. (approx. \$20K)
- Provide clinician for a school that does not have the skills yet.

- SANS to fund 4 schools
- SAFECODE joining in the initiative (funding and providing clinicians)
- Will you help?



**The Most Trusted Name in
Information Security**





HOW CAN YOU MEASURE SOFTWARE DEVELOPMENT IMPROVEMENT?



The Most Trusted Name in
Information Security



The Top 25 CWEs

- If we know which weaknesses are most important, and how important they are, then measuring their reduction is a rational approach to measuring effectiveness of software development process improvements

How To Define the Top 25 CWEs

- Get help from an extraordinary team:
 - Bob Martin and Steve Chistey from Mitre
 - GSSP Blueprint Team (criticality, importance, frequency):
 - Matt Bishop, UC Davis
 - Sanjay Bahl, TATA/Microsoft India
 - Chris Telfer,
 - Ryan Berg, Ounce
 - Ed Tracy, BAH
 - Randy Marchany and Ruilang Chen, VT

How You Can Help

- First draft is expected by the end of the week.
- If you know a lot about secure programming errors and are willing to help with the selection and prioritization, we need your help!
- Project organization:
 - Use 'big vulnerability' or 'class' as a way to organize; then we focus on the actual error...which is actionable (especially when we get to language specific stuff some day).

Questions

- Secure coding clinics
- Top 25 CWEs