Discovering Software Vulnerabilities through Interactive Static Analysis

http://aside.uncc.edu/

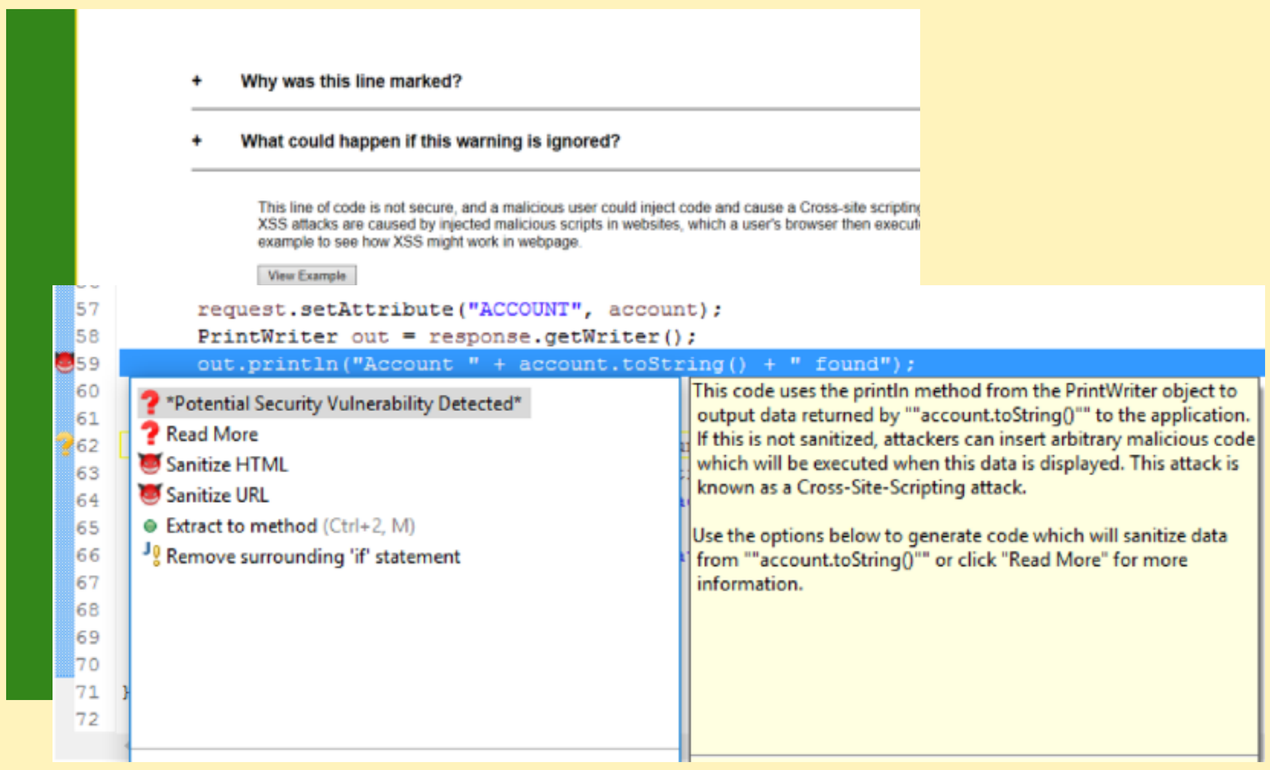
Heather Lipford, Bill Chu: University of North Carolina at Charlotte

Emerson Murphy-Hill: North Carolina State University

Software vulnerabilities originating from insecure code are a leading cause of security problems. We propose a new approach, **interactive static analysis**, to integrate the detection and mitigation of security vulnerabilities into the context of development.

Problem

- Static and dynamic security tools require **significant knowledge** to use and understand;
- Such tools tend to be run at the end of development by **security experts**
- Thus, developers are kept "out of the security loop", and continue to introduce security vulnerabilities.



ASIDE: Application Security in the IDE

Prototype interactive static analysis tool

Approach

- Understand how developers interpret and diagnose security warnings
- Integrate static analysis within the IDE, increasing developers' **awareness and practice of secure programming**
- Help developers resolve vulnerabilities through **automated code generation** and providing sufficient explanations;
- Utilize the programmer's **contextual knowledge** to drive customized static analysis, detecting additional vulnerabilities.

Tool Development and Evaluation

ASIDE is an Eclipse plug-in for Java and PHP that detects and provides security vulnerability warnings alongside code in the IDE.

- **Automated code generation** of sanitization of untrusted input and encoding of output;
- **Interactive annotation** of application-specific security decisions, driving detection of access control and CSRF vulnerabilities.

Evaluation on one large open source project (Moodle) and multiple smaller projects:

- 3 zero day vulnerabilities detected
- Better coverage of known vulnerabilities over other static analysis methods

Multiple user studies of advanced students and developers demonstrates that ASIDE is **usable**, and **increases user awareness** and **knowledge** of security vulnerabilities.

Studying existing security tools

- Studied how developers diagnose vulnerabilities in an open source project using Find Security Bugs.
- Analyzed the questions developers asked, including questions surrounding the vulnerabilities and attacks, but also the software, related resources, and tools.
- Analyzed the strategies they used to answer questions, and the success and failure of those strategies.

Expanding beyond the IDE

Study of 30 security professionals who perform security analysis of code, enhancing our understanding of the life cycle of code analysis.

Additional detection and mitigation of vulnerabilities:

- Automatic unit testing of sanitization functions
- Using interactive static analysis to support security-oriented code review by developers

Interested in meeting the PIs? Attach post-it note below!

National Science Foundation
WHERE DISCOVERIES BEGIN

The 3rd NSF Secure and Trustworthy Cyberspace Principal Investigator Meeting
January 9-11, 2017
Arlington, Virginia

NC STATE UNIVERSITY
UNC CHARLOTTE