CS4239
Software Security (2015/2016, Semester 1)

Lesson Plan : Software Security

Updated: 07-Apr-2015

Week 1: 10 Aug-14 Aug

Introduction - what the module is about, discussing background for the module, specifically any systems background or mathematical background needed for the module.

Background - program representations

Week 2: 17 Aug-21 Aug

Buffer overflow attacks

Week 3: 24 Aug-28 Aug

Summary of software vulnerabilities - specifically SQL injection.

Lab: Introduction to LLVM

Week 4: 31 Aug-04 Sep

Static analysis - an introduction. Static dependency analysis.

Static analysis for detection of software vulnerabilities

[sample paper http://suif.stanford.edu/papers/usenixsec05.pdf]

Lab: elaboration of such analysis using LLVM

Week 5: 07 Sep-11 Sep

Begin general discussion on dynamic analysis, including dynamic symbolic execution

Lab: More in-depth study of LLVM, including its IR.

Week 6: 14 Sep-18 Sep

Dynamic analysis - general introduction plus dynamic symbolic execution

Dynamic analysis to find software vulnerabilities

[sample paper: OSDI 2008 paper of KLEE, FSE 2010 paper from NUS]

Lab: Introduction to KLEE

Week 7: 28 Sep-02 Oct

Midterm Examination

+

Lab: In-depth use of KLEE for bug-hunting

Week 8: 05 Oct-09 Oct

Digging deeper: **Implementing static and dynamic taint analysis**
(to recommend related papers - can read Dytan paper from ISSTA 2007, and the references within)

Lab hour: Look into implementing tainting capabilities inside LLVM (can be project deliverable 1)

Week 9: 12 Oct-16 Oct

Digging deeper: Difference between dynamic symbolic and concolic executions
[Can read the papers: DART, CUTE etc in this class, also possibly the ISSTA 2011 paper by Visser]

Lab: Implementing a DSE engine - learning from KLEE (some fragments will be given to the students, and for a restricted subset of instructions)

Week 10: 19 Oct-23 Oct

Software model checking using symbolic execuction and its usage in vulenrability detection

Lab: Continue with the implemenation of DSE in LLVM -> project deliverable 2

Week 11: 26 Oct-30 Oct

Black-box fuzzing and related testing issues.

Lab: Symbolic JPF, look inside its implementation [project deliverable 3 ?]

Week 12: 02 Nov-06 Nov

Crypto vulnerabilities

Lab: open discussion among students on the various tools shown - LLVM, KLEE and JPF
- combination of project deliverables need to think about this - since JPF is for Java programs

Week 13: 09 Nov-13 Nov

Sample emerging topic: Program patching [papers by Matthias Payer, plus works by Mckinley, plus recent works on program repair including works from NUS etc]

Lab: Flexible.