Software Reliability and Security

Module 1

Winter 2017

Objective and Audience

- Objective
  - Omnipresence of software controlled systems - quality of service (QoS) is extremely important
  - To assure QOS - need for reliable and secure software is obvious
  - Some of the underlying principles and the state-of-the-art research on software reliability and security topics
- Audience
  - Graduate students with general background in computer science, computer engineering, or electrical engineering
  - Some programming experience is required

About the Instructor

- Canada Research Chair in Software Dependability
- Professor, Queen's School of Computing, Electrical & Computer Engineering (cross-appointed)
- Teaching: Software quality assurance, Distributed systems, Software reliability and security
- Research: Software Dependability - methods and tools for reliable and secure software
  - Queen's Reliable Software Technology (QRST) Group
- Primary Contact: 535 Goodwin Hall, mzulker@cs.queensu.ca
- More information: http://cs.queensu.ca/~mzulker

Introduction

- General Course Information
  - Overview and some preliminary concepts
  - Lecture schedule
  - Topics and references
  - Assessment with report due dates
  - Warm-up presentation

Trustworthy/Dependable Computing

Trustworthy computing is computing that is as available, reliable and secure as electricity, water services and telephony.

-Bill Gates (January 2002, Washington)

Dependability computing is about the trustworthiness of a computer system such that reliance can be justifiably be placed on the service it delivers.

-IFIP Working Group 10.4 (IEEE 1985)

Dependability/Quality

- Computer Dependability [IEEE 1985]
  - Available – ready to use when needed
  - Reliable - continuity of service
  - Safe – avoid catastrophic consequences
  - Secure - unauthorized access of information

- Software Quality Equation [Voas 2002]
  - Q = aR+bP+cF+dSa+eSe+fA+gM+hT
  - Reliability, performance, fault-tolerance, safety, security, availability maintainability, testability, …

- Trustworthy software?

Trustworthy – it is complicated!

source unknown

- Reliable software - free from software failures while operating
- Secure software - function properly under malicious attacks

Software Reliability and Security

- Software Reliability
  - Probability of failure-free software operation for a specified period of time in a specified environment
  - Related areas – Fault/failure prevention, detection, removal, tolerance, and forecasting
- Software Security
  - Confidentiality, integrity, and availability of a software system with respect to some policies
  - Related areas – risk management, secure design, programming languages and environments, auditing, vulnerability analysis and testing

Software Reliability Engineering

- Software Engineering (IEEE standard 610.12-1990)
  - The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; …
  - Related areas – Usability, performance, timely completion, maintenance
- Software Reliability Engineering [IEEE95]
  - The quantitative study of the operational behavior of software-based systems with respect to user requirements concerning reliability
  - Related areas – Reliability measurement, reliability attributes in software development and operation

Software Security Engineering

- Security Engineering [Anderson 2001]
  - About building systems to remain dependable in the face of malice, error, or mischance, …
  - Related areas – Access control and authentication, confidentiality, integrity, intrusion detection, trust and privacy

- Software Security Engineering [McGraw 2001]
  - Developing secure software – Develop software systems that meet both security and software functionality objectives
  - Related technical areas - risk management, secure design, programming languages and environments, auditing, vulnerability analysis and testing

Topics and References

- Topics (subject to minor changes)
  - Software Crisis
  - Software Process Models
  - Software Reliability
  - Methods for Reliable Software
  - Software Reliability Engineering Process
  - Software Dependability
  - Software Fault Tolerance
  - Software Security
  - Program Security
  - Software Security Engineering Process
- References
  - Lecture notes, book chapters, journal and conference papers
  - Details will be provided after each lecture

Topics – contd.

- Software Crisis
  - Delivered software products are not reliable or secure
- Software Process Models
  - A number of steps/tasks for developing software
- Software Reliability
  - Probability of failure-free software operation for a specified period of time in a specified environment
- Methods for Reliable Software
  - Apply different but complementary techniques to the artifacts that appear throughout the software life
  - Formal methods, testing, inspection, fault-tolerant computing

Topics – contd.

- Software Reliability Engineering Process
  - Define necessary reliability
  - Develop operational profiles
  - Prepare for test
  - Execute test
  - Apply failure data to guide decisions

- Software Dependability
  - A broader concept to include safety, reliability, security, and availability

- Software Fault Tolerance
  - A system is considered fault tolerant if the behavior of the system, despite the failure of some of its components, is consistent w.r.t. its requirements
  - Phases: Error detection, Damage confinement and assessment, Error recovery, Fault treatment and continued service

Topics – contd.

- Software Security
  - Software security is part of computer security emphasizing on software
  - Confidentiality, integrity, and availability (a.k.a. CIA) of a software system with respect to some policies
- Software Security Engineering Process
  - Develops a software system that remains operational even when it is under an attack
  - Emphasizes on the methods and tools to specify, design, implement, and test secure software systems

Lecture Schedule

- Lecture Hours
  - Wednesdays and Fridays 10:00 am – 11:30 am, Goodwin Hall 521
- Course Website
  - http://www.cs.queensu.ca/~mzulker/cisc848.html
- Academic Integrity
  - http://www.queensu.ca/artsci/academics/undergraduate/academic-integrity
- Use of Lecture Notes
  - The lecture notes are taken from various copyrighted sources for the sole classroom use of the students registered in CISC 848
  - Do no disseminate the lecture notes in any form
  - Always refer to the original source – not these lecture notes

Assessment – Marking Scheme

- Presentation 40%
  - 3 Presentations & Class Participation – 4X10
- Reports and Project (Group of 1-3) 40%
  - Proposal Report – 10
  - Final Report – 10
  - Project Work – 20
- Final Exam 20%
- Audit Requirements
  - 33% of the total course work
  - A combination of presentation, class participation, project
- No Incomplete Grade

Presentations

- Warm-up (background) presentation
  - May be related to your project
- Two presentations on your project
  - Proposal and final project
- Provide presentation slides before your presentation
- Presentation length
  - 20 minutes

Presentation/Lecture Schedule and Report Due Dates

- Presentation 1
  - Related background paper
  - Jan 27, Feb 1, 3
- Presentation 2
  - Project proposal
  - March 1, 3, 8
- Presentation 3
  - Final project report
  - March 24, 29, 31

- Lectures

  Jan 13, 18, 20, 25, 27

  Feb 1, 3, 8, 10, 15, 17

  March 1, 3, 8, 10, 15, 17, 22, 24, 29, 31

- Project Proposal Due
  Tuesday, February 28

- Final Project Report Due
  Monday, April 10

- Final Exam
  Wednesday, April 12, 10:00am

Warm-Up Presentations

- Presentations will start on January 27
- Summary of at least two full conference/journal papers published in 2012-2016
- You can discuss with me for selecting papers after the lectures
- How the papers are related to each other?
- How the general concept of the papers related to the course topics?
- You can choose papers thinking about your project (But it is ok if it is not related to the project you are thinking of)

Warm-Up Presentation - contd.

- Answer the following in your presentation
  - Main motivation?
  - Problems/contributions?
  - Solution approach ?
  - Conclusions /lessons learned?
  - Future work?
- General Advice
  - Provide the paper and the slides to me before your presentation
  - Try to use your own examples in the presentation
  - Think about the audience in the class so that they can understand
- An interesting reference
  - http://www.acsac.org/speakers.pdf

Summary

- General Course Information
  - Overview and some preliminary concepts
  - Lecture schedule
  - Topics and references
  - Assessment with report due dates
  - Warm-up presentation
- Next
  - Software Crisis
    - Reliability and security
    - An assignment – recent incidents about software failures and intrusions
  - Course Project
    - Guidelines
    - Some suggested topics

Some Lecture Sources

- W. Gibbs, Software's Chronic Crisis, Scientific American, pages 86-95, September 1994
- Jeffrey Voas, "Trusted Computing's Holy Grail," DSN 2002.
- J. Viega and G. McGraw, "Building Secure Software: How to Avoid Security Problems the Right Way," Addison-Wesley Pub Co, 2001
- M. Dowd, J. McDonald, and J. Schuh, *The Art of Software Security Assessment*, Addision-Wesley publications, 2007.
- R. Anderson, "Security Engineering - A Guide to Building Dependable Distributed Systems," Wiley, January 2001