USC Informatics Program 523 (INF 523): Assurance in Cyberspace - Fall 2018

Lecture Friday - 1PM to 4:20 PM in OHE 120
Clifford Neuman

Announcements

Schedule

- First Lecture - 24 August 2018, 1PM - 4:20 PM in OHE 120
- Mid-term exam - Friday October 12, 2018 - 1PM - 3PM followed by lecture
- Final Exam - Wednesday December 12, 2018 - 11 AM to 1 PM

Course Materials

Homework Assignments

- Assignment #1 - Due 14 Spetember 2018

Lecture Slides

- Slides for Lecture 1-3 - ppt
- Lecture Videos and Discussion forum at DEN D2L

Course Summary

Assurance as the basis for believing an information system will behave as expected. Approaches to assurance for fielding secure information systems that are fit for purpose. Recommended preparation: Prior degree in computer science, electrical engineering, computer engineering, management information systems, and/or mathematics. Some background in computer security preferred.

The definition of security for a system is given by the security policy. A system is “secure” only insofar as it correctly implements the security policy. But flaws in a system’s design and implementation may create vulnerabilities that allow an attacker to violate that policy, and the complexity of computer systems make it difficult to verify that a system’s design and implementation are free of flaws. In fact, the current state-of-the-art in system development is incapable of “proving” that a system of more than trivial complexity is secure.

Because absolute proof about the security of a system is (at least with current technology) unobtainable, a system’s “assurance case” – the argument that the system correctly implements the security policy – is formed from a body of supporting evidence generated at different stages of the system lifecycle. This course will explore different techniques and methods for creating the assurance case.

Students will develop the following abilities

- To identify stages in the system lifecycle where flaws may be introduced into a system
- To describe methods and techniques for creating an assurance argument
- To evaluate the strengths and weaknesses in each of the methods and techniques
- To balance costs with benefits of applying each method and technique based on an assessment of risk
- To create an assurance argument using the body of evidence generated at different stages in a system's lifecycle

Instructors and Assistants

Clifford Neuman

- Office: Information Sciences Institute - 310-448-8736
- Office hours: Friday 12:20-12:50 and 4:30-5:00 PM - Office RTH 512 - or by appointment
- Email: inf523 at csclass.info (to Instructor and TAs)

Reading Assignnments

Week 1: Introduction to Assurance - Before Lecture on August 31

- Chapter 18, "Introduction to Assurance"
  – Computer Security Art and Science: Bishop, Matt, 2003
- Introduction tothe Secure Software Development Lifecycle

Week 2: Measuring Security - Before Lecture on 31 August 2018

- ISACA-How Can Security Be Measured?
- ISACA-Performing a Security Risk Assessment

Covered in Week 2, but read before Lecture on 7 September

- TCSEC, pp. 10, 50-53, 62-63, 67-68, 77-79
- Common Criteria, Part 3, pp. 15-17, 44-45
- Final Evaluation Report, Gemini Trusted Network Processor – Section 7
- SSE-CMM/ISO 21827 Capability Maturity Model (and searche for "ISO 21827")
- Build Security In Maturity Model (BSIMM) - (website)
- Microsoft Security Development Lifecycle

Week 3 - Attack and Fault Modeling - Read Before Lecture on 7 September

- Attack Trees
- A Requires/Provides Model for Computer Attacks
- Uncover Security Design Flaws Using the STRIDE Approach

- Foundations of Attack–Defense Trees
- Threat Risk Analysis for Cloud Security based on Attack-Defense Trees

Week 4 - Structured Design - Read Before Lecture on 14 September

- D. L. Parnas, On the Criteria To Be Used in Decomposing Systems into Modules, 1972
- Daniel Hoffman, On Criteria for Module Interfaces, 1990
- Paul Karger, et. al. A VMM security kernel for the VAX architecture, 1990 – Section 3.7
- Final Evaluation Report, Gemini Trusted Network Processor, 1995 – Section 4.2

Week 5 - Secure Programming - Read Before Lecture on 21 September

-
- IEEE – Avoiding the top 10 Software Security Design Flaws
- [Skim] CERT Top 10 Secure Coding Practices
- [Skim] Common Weakness Enumeration
- [Skim] OWASP Top 10 2013
- ***NEW*** To Type or Not to Type: Quantifying Detectable Bugs in JavaScript/a>

Week 5 - Testing - Read Before Lecture on 21 September

- Analysis Techniques for Information Security, pp. 5-10 (static testing)
- Nathaniel Ayewah, David Hovemeyer, J. David Morgenthaler, John Penix, William Pugh, Using static analysis to find bugs, IEEE Software, vol. 25, no. 5, pp. 22–29, Sep./Oct. 2008
- P. Oehlert, Violating assumptions with fuzzing, 2005 (fuzzing/dynamic testing)
- Jose Fonseca, et. al., Testing and comparing web vulnerability scanning tools for SQL injection and XSS attacks, 2007 (vulnerability scanning)

Week 6 - Presentation Proposals - Individual Readings

Week 7 - Vulnerability Scannig and Pen Testing

Bishop book, Chapter 23, "Vulnerability Analysis", pp. 645-660 (penetration testing)

- Implementation of Tripwire: A File System Integrity Checker, Gene Kim, 1993.

Week 8 - Mid-term Examp - 12 October 2018

Week 9 - Formal Methods Introduction - 19 October 2018

- Bishop, pp. 545-551
- A Specifier's Introduction to Formal Methods, Jeannette M. Wing

http://www.cs.cmu.edu/~wing/publications/CMU-CS-90-136.pdf

- Formal Specifications, a Roadmap, Axel van Lamsweerde (Also in D2L)

  http://www.ufpa.br/cdesouza/teaching/es/finalvanlamsweerde.pdf

- Jonathan K. Millen. 1976. Security Kernel validation in practice. Comm. ACM 19, 5 (May 1976), 243-250. DOI=10.1145/360051.360059

  http://dl.acm.org/citation.cfm?id=360059

Week 10 - Covert Channels - 26 October 2018

- Bishop book, Chapter 17 Confinement Problem

Shared Resource Matrix Methodology: An Approach to Identifying Storage and Timing Channels, Richard Kemmerer, 1983

Covert Flow Trees: A Visual Approach to Analyzing Covert Storage Channels, Richard Kemmerer, 1991

An Entropy-Based Approach to Detecting Covert Timing Channels, Steven Gianvecchio and Haining Wang, 2011

Week 11 - Secrity Kernel Case Studies - 2 November 2018

- T. Levin, S. Padilla, and R. Schell, Engineering Results from the A1 Formal Verification Process, in Proceedings of the 12th National Computer Security Conference, Baltimore, Maryland, 1989. pp. 65-74

  http://csrc.nist.gov/publications/history/nissc/1989-12th-NCSC-proceedings.pdf

- A Multilevel File System for High Assurance

  http://calhoun.nps.edu/bitstream/handle/10945/7177/95paper_mls.pdf?sequence=3

Week 12 - File System and Higer Levels Case Studies - 9 November 2018

- A Multi-Level Secure File Sharing Server and its Application to a Multi-Level Secure Cloud

  http://mrheckman.com/yahoo_site_admin/assets/docs/Aesec-MILCOM15-FileServer-151028-COPYRIGHT.300141512.pdf

Week 13 - Subversion Case Studies - Network Guard Case Studies - 16 November 2018

- A Demonstration of the subversion threat: facing a critical responsibility in the defense of cyberspace

  http://calhoun.nps.edu/bitstream/handle/10945/6073/02Mar_AndersonE.pdf?sequence=3&isAllowed=y

- A High-assurance, Virtual Guard Architecture

  http://ccss.usc.edu/INF527/fall15/IEEE-MILCOM12-Heckman-Guard.pdf

- GemSeal Guard- High Assurance MLS

  http://ccss.usc.edu/INF527/fall15/GemSeal-Guard-070117.pdf

- Recon Guard Report

  http://ccss.usc.edu/INF527/fall15/RECONGuardReportacrov5.pdf

Course Grade Components

A letter grade will be assigned for each assignment, project, or exam. The individual assignment, project, and exam scores are based on student performance relative to other students in the class. The final course grade will be determined by weighted calculation from the component grades, and may be adjusted upward if the students participation is exemplary. The components of the final course grade are:

- Mid-Term Exam 20%
- Final Exam 20%
- Class Participation 10%
- Homework and/or Quizzes 25%
- Case Study 25%

Academic Integrity

As an instructor I take academic integrity seriously. Cases of academic misconduct will result in the assignment of a failing grade for the class and referal of the matter to the student conduct office. In each of the past several years I have turned in multiple students for cheating and assigned failing grades. Information on what constitutes academic dishonesty can be found on the CSci530 academic integrity page, and by following links to university resources found on that page.

Exams from Prior years

- Spring 2016 Mid-term Exam
- Spring 2016 Final Exam