INF 525: TRUSTED SYSTEM DESIGN, ANALYSIS, AND DEVELOPMENT
4 units

Spring 2018 Syllabus

Instructor	Email	Office	Office Hours	Lecture
Tatyana Ryutov	tryutov@usc.edu	PHE 336	TBD	Mon 2:00-5:20pm OHE 100D

Course Description

High consequence applications such as those for critical infrastructure require highly reliable, trusted systems to assure the required availability of processing, and to assure the required confidentiality and integrity of information and processing, even if some parts of the system have high exposure to a witted adversary employing subversion. Hardware and software design techniques for these Trusted Systems must evolve and advance as the sophistication of the cyber adversary also advances. This course conveys a methodology for the development of trusted systems using the Reference Monitor concept as a unifying principle. Highly secure Trusted Systems are based on what is called a Security Kernel that incorporates the Reference Validation Mechanism – the hardware and software that implements the Reference Monitor.

Trusted Systems lay at the core of secure systems. A detailed understanding of the design, analysis and implementation of trusted systems is essential for the development of secure information systems. This course provides an overview of computer security to include an analysis of what is computer security, why systems are not secure, and the general concepts and design techniques applicable to the design of hardware and software. It examines in detail the principles of a security architecture, access control, policy and the threat of malicious code; the considerations of trusted system implementation to include hardware security mechanisms, security models, security kernels, and architectural alternatives; the related assurance measures associated with trusted systems to include documentation, formal specification, verification, and testing. That core needs to be sufficiently capable that it can be leveraged by approaches that extend the trusted system, into applications such as databases and into networks and distributed systems.

This class will be primarily individual study, with weekly assigned readings, homework assignments, one semester project, a midterm examination and a final examination.

Course Resources

Piazza will be used for lectures, announcements, assignments, and intra-class communication

- DEN D2L will be used for:
- posting of grades
- homework submission
- quiz submission (DEN student only)

Prerequisite(s): INF 519 – Foundations and Policy for Information Security

Learning Objectives

Students will have ten learning objectives for the course, and additional technology application objectives.

Learning Objectives:

1. Understand the fundamental issues that motive computer security to include the impediments and the motivating threat strategies such as subversion
2. Understand the technical basis for the development of trust in computer systems
3. Understand the relationship between trust and policy in trusted computer systems, and the pivotal role of a formal security policy model
4. Understand in depth the techniques and approaches for designing trusted technology in computer systems, including information hiding and layering
5. Understand the relationship and dependences between the underlying hardware and the trusted technologies that can be built on that hardware
6. Understand and be able to apply the fundamental design considerations for trusted systems
7. Understand in detail the concepts of the reference monitor and the nature of the root of trust provided by cryptographic attestation.
8. Understand the architectural issues that are essential to the implementation of trusted technology, including implications of hardware segmentation
9. Understand the processes for specification of trusted systems and how that specification relates to the sufficiency of trusted technology
10. Understand the extension of the trust model into trusted applications

Technology Application Objectives:

1. Synchronization
2. System initialization
3. Protection rings
4. Multiprocessing
5. Virtualization
6. Non-discretionary security representation generality
7. Trusted Distribution
8. Hardware root of trust
9. Methods of analysis and evaluation
10. Software engineering practices of secure system development and management

Grading

Grading method will be relative and on the curve.

Technological Proficiency and Hardware/Software Required

This course is intended for graduate students typically coming out of computer science, mathematics, computer engineering, or informatics. Students need to be familiar with operating system principles and able to program. Advanced knowledge of computer architecture, theory of computation, and communications networks will be valuable. Students should be thoroughly familiar with the reference monitor abstraction of system security, as well as with the associated common mathematical models and techniques for their implementation, interpretation and objective evaluation.

Required Readings and Supplementary Materials

All books, papers or reports will be available to students in one of three ways: 1) in the USC bookstore or other commercial source; 2) via Course Documents that the instructor will provide on Blackboard; and/or 3) via the web including USC library access.

Required Readings:

TEXTBOOKS:
[GAS] Building A Secure Computer System, by Morrie Gasser, Van Nostrand Reinhold, New York, 1988.

[PFL] Security in Computing, Pfleeger, Prentice-Hall, 4th ed. 2006.

[ORG] The Multics System: An Examination of Its Structure, by Elliot L. Organick, The MIT Press, Cambridge, Massachusetts, 1972.

[OSS] Operating System Security, Trent Jaeger, 2008.

LITERATURE:
[A1M] “Security Requirements for a Class A1 M-Component”, Extracts from Trusted Network Interpretation. "NCSC-TG 005." National Computer Security Center (1990), prepared August 17, 2005.

[AMES] Ames Jr, Stanley R., Morrie Gasser, and Roger R. Schell. "Security kernel design and implementation: An introduction." Computer 16.7 (1983): 14-22.

[CRO] Alexander Crowell, Beng Heng Ng, Earlence Fernandes, Atul Prakash: The Confinement Problem: 40 Years Later. JIPS 9(2): 189-204 (2013).

[EVC] Reed, David P., and Rajendra K. Kanodia. "Synchronization with eventcounts and sequencers." Communications of the ACM 22, no. 2 (1979): 115-123.

[GKS] Schell, Roger, Tien F. Tao, and Mark Heckman. "Designing the GEMSOS security kernel for security and performance." Proceedings of the 8th National Computer Security Conference. Vol. 30. 1985.

[EPL] Evaluated Product List, Gemini Computers, Incorporated, GTNP Version 1.01, Network Component, M Only, CSC-EPL-94-008, National Security Agency, 6 September 1994.

[LEV] Levin, T. E., Tao, A., & Padilla, S. J. (1990). Covert Storage Channel Analysis: A Worked Example. Proc. National Computer Security Conference, 10-19.

[MTS] Schell, R.R., and Tao, T.F., Microcomputer-Based Trusted Systems for Communication and Workstation Applications, Proceedings of the 7th DoD/NBS Computer Security Initiative Conference, NBS, Gaithersburg, MD, 24-26 September 1984, pp. 277-290.

[**MULT**] Bensoussan, Andre, Charles T. Clingen, and Robert C. Daley. "The Multics virtual memory: concepts and design." Communications of the ACM 15.5 (1972): 308-318.

[**FER**] Final Evaluation Report, Gemini Computers, Incorporated, Gemini Trusted Network Processor, Version 1.01, National Computer Security Center, 1995.

[SAL] Saltzer, Jerome, and Schroeder, 1975. "The protection of information in computer systems," Proc IEEE 63(9), September, 1975.

[SFG1] GTNP Security Features User's Guide, Vol 1, Introduction to the GEMSOS Security Kernel, GNT00-SFG01-0005C, April 24, 2003.

[SFG2] GTNP Security Features User's Guide, Vol 2, Programmer's Guide to the GEMSOS Security Kernel Interface, GTN00-SFGP2-0008a, June 1, 2004.

[SCH] Schiller, W. L., The Design and Specification of a Security Kernel for the PDP-11/45 (1975).

[TCSEC] Department of Defense, 1985, Department of Defense Trusted Computer System Evaluation Criteria, DoD 5200.28-STD, Washington, DC.

[VMM] Karger, Paul A., Mary Ellen Zurko, Douglas W. Bonin, Andrew H. Mason, and Clifford E. Kahn. "A retrospective on the VAX VMM security kernel." Software Engineering, IEEE Transactions on 17, no. 11 (1991): 1147-1165

[STACK] Smashing the stack for fun and profit, by Aleph One.

[BUF] Buffer Overflows: Attacks and Defenses for the Vulnerability of the Decade by Crispin Cowan, Perry Wagle, Calton Pu, Steve Beattie, and Jonathan Walpole.

[LOW] Low-level Software Security by Example by Ulfar Erlingsson, Yves Younan, and Frank Piessen.

[RET] Return-Oriented Programming: Systems, Languages, and Applications by Ryan Roemer, Erik Buchanan, Hovav Shacham, and Stefan Savage.

Supplemental Readings:

[MCGRAW] Software Security: Building Security In, book by Gary McGraw.

[HW] Introduction to Hardware, Security and Trust, book by Mohammad Tehranipoor, Cliff Wang, 2012.

Description and Assessment of Assignments

Students will be required to complete several homework assignments, which may take several hours to complete. All homework assignments are to be prepared and submitted individually;

however students may work in groups to understand and discuss the tasks. There is one midterm test and a final exam which date will be determined by the College. There will be several short in-class quizzes. There will be several homework assignments and one semester project.

In class exams and quizzes will be closed book, no notes, no crib-sheets, no electronic devices. Exams/quizzes missed due to a verified serious illness will be assigned a grade scaled from other work.

Guidelines and additional information will be developed and provide for the submission of assignments.

The following are representative of the homework assignments to be completed by students. The details of the assignment will vary from semester to semester.

Sample assignment 1
The main purpose of this assignment is to develop the initialization for an information-hiding (Parnas) module. For this development you are to exercise specific best software engineering practices that are beneficial for developing a trusted system. Some examples of these best practices are creations of data structures that reflect information hiding, no storage of pointer types in your data structure, separation of input and output parameters, and use of pass by value for parameter passing.
The objective of the programming assignment is to provide the opportunity to practice software engineering techniques as you study them in this course.

Sample assignment 2
This assignment builds directly on and extends the previous "Assignment 1" the main purpose of which was to develop the initialization for an information-hiding (Parnas) module. Assignment 1 requirements apply to this assignment, even though they are not reiterated here. For the current assignment you will modify the code from that assignment by adding additional entry points to create an updated module for managing an abstract list of records. Your new updated module must fully meet all the abstract specification for Assignment 1, as well as all its required best software engineering practices, even though they are not repeated in this assignment.

Semester Project

The semester project gives each student the opportunity to use and illustrate the concepts from the course in an applied manner. In not less than 7 no more than 15 pages prepare a report in Word or PDF format with a font size between 10 and 12. Figures, tables, and the like are not included in the 15 page maximum page count, but text beyond the 15 page limit will not be considered in grading. Submit the report in electronic form on USC blackboard. This course conveys a methodology for the development of trusted systems using the Reference Monitor concept as a unifying principle.

This first security kernel was (1) informed by a quite early version of a Formal Security Policy Model (FSPM), (2) built long before there was any concrete evaluation criteria (e.g., the TCSEC and TNI), and (3) constructed without the benefits of significant research results that came later. Based on information you gather and review, your report is to reflect your analysis of the suitability of this early proof of concept to verifiably enforce a security policy. The scope of this project will be limited to the minimum set of policy requirement to address RVM capabilities focused on MAC policy. To be definitive, this project is to focus the requirements for TNI Mandatory Only Components (M-Components), as codified in TNI Sections 4.1.1 (Policy) and A.3.1 (M-Comp). The associated RVM evaluation factors have been identified, and for reference these requirements are each detailed in extracts from the TNI.

The design for this project is available in the literature. Based on information you gather and review, you are to report your research and analysis of how this early proof of concept developed by Lee Schiller did, and did not, satisfy requirements of the design and development technologies we have studied.

Assignment Submission Policy

Students may work in groups to complete homework. All assignments are to be submitted individually.
Assignments and projects are due on time. There is a substantial grade penalty for late submission. Cumulative of 10% times number of days late:

- 1 day late: lose 10%
- 2 days late: lose 30% (10% + 20%)
- 3 days late: lose 60% (30% + 30%)
- Greater than 4 days late not accepted

Additional Policies

Diversity

The diversity of the participants in this course is a valuable source of ideas, problem solving strategies, and engineering creativity. I encourage and support the efforts of all of our students to contribute freely and enthusiastically. We are members of an academic community where it is our shared responsibility to cultivate a climate where all students and individuals are valued and where both they and their ideas are treated with respect, regardless of their differences, visible or invisible.

Students with Disabilities

Any student requesting academic accommodations based on a disability is required to register with Disability Services and Programs (DSP) each semester. A letter of verification for approved accommodations can be obtained from DSP. Please be sure the letter is delivered to me as early in the semester as possible. Your letter must be specific as to the nature of any accommodations granted. DSP is located in STU 301 and is open 8:30 am to 5:30 pm, Monday through Friday. The telephone number for DSP is (213) 740-0776.

Return of Course Assignments

Returned paperwork, unclaimed by a student, will be discarded after a year and hence, will not be available should a grade appeal be pursued following receipt of his/her grade.

Projected Course Schedule: A Weekly Breakdown

Class sequence, dates, reading assignments, and topics are subject to change as the semester proceeds. Any revisions will be noted and announced in class.

	Topics	Readings and Homework	Deliverable
Week 1 1/08	Course introduction, Overview of Trusted Operating System Design and Security Kernel (SK) Approach, Software engineering practices of secure system development and management	GAS Chapters 1-5 PFL Chapters 5.1, 5.4 AMES GAS Chapters 6, 8 EPL **Homework 1 assigned**	
Week 2 1/15	**NO Class: Martin Luther King Day, University holiday**		
Week 3 1/22	Design of SK modules: information hiding, layering and minimization Reference monitor objects implemented by SK as segmentation	GKS MTS SFG1 Sections 1-4 SFG2 Sections 4.1 and 4.6.13 **Quiz 1**	
Week 4 1/29	Security kernel layering Designing a security kernel Detailed design principles for secure systems	GAS Chapters 9, 10 ORG Section 1.2, pages 1-13 GKS LEV **Homework 2 assigned**	**Homework 1 due**
Week 5 2/5	Trusted system building techniques Trusted path, Trusted functions	SFG1 **Quiz 2**	
Week 6 2/12	Kernel implementation strategies Confinement and covert channels	CRO GAS Chapter 7	
Week 7 2/19	**NO Class: Presidents' Day, University holiday**		
Week 8 2/26	Covert channel analysis Synchronization in a trusted system	FER pages 61-63 VMM FER Sections: 8.6, 8.11, 10.6 **Quiz 3**	**Homework 2 due**
Week 9 3/5	**Midterm** **Topic TBD**		
	Spring Break		
Week 10 3/19	Secure initialization and configuration Low-level, memory-based attacks and defenses Secure Software Development	EVC FER Sections: 4.2.1.6, 4.2.1.7.3, 6.1, 6.1.1, 8.10, 10.5 STACK, BUF, LOW	

Week 11 3/26	Static Analysis, Symbolic Execution, Penetration Testing, Fuzzing	RET MCGRAW (supplemental) **Homework3 assigned** **Quiz 4**	
Week 12 4/2	Trusted computing, Trusted Platform Module Trusted distribution	FER Sections: 2.4, 4.0, 4.1.9, 4.2.1.3.5, 4.2.1.8.2.6, 4.5.1, 4.5.2, 7.8, 8.20, 8.22	
Week 13 4/9	Capability systems and separation kernels Management of SK rings and labels, SK as system enabler	OSS Chapters 10 and 11.1 FER Sections: 4.2.1.6, 4.2.1.7.3, 6.1, 6.1.1, 8.10, 10.5 SFG1 Sections: 6-6.4 **Quiz 5**	
Week 14 4/16	Hardware security, vulnerabilities, attacks, Hardware Trojans Techniques for building secure and trusted hardware	HW (supplemental)	**Homework 3 due**
Week 15 4/23	Security analysis of trusted systems, Course review	A1M **Quiz 6**	**Semester project due**
FINAL EXAM	**Monday, May 7, 2-4 p.m.**		

Statement on Academic Conduct and Support Systems

Academic Conduct

Plagiarism – presenting someone else's ideas as your own, either verbatim or recast in your own words – is a serious academic offense with serious consequences. Please familiarize yourself with the discussion of plagiarism in *SCampus* in Section 11, *Behavior Violating University Standards* https://scampus.usc.edu/1100-behavior-violating-university-standards-and-appropriate-sanctions. Other forms of academic dishonesty are equally unacceptable. See additional information in *SCampus* and university policies on scientific misconduct, http://policy.usc.edu/scientific-misconduct.

Discrimination, sexual assault, and harassment are not tolerated by the university. You are encouraged to report any incidents to the *Office of Equity and Diversity* http://equity.usc.edu or to the *Department of Public Safety* http://adminopsnet.usc.edu/department/department-public-safety. This is important for the safety of the whole USC community. Another member of the university community – such as a friend, classmate, advisor, or faculty member – can help initiate the report, or can initiate the report on behalf of another person. *The Center for Women and Men* http://www.usc.edu/student-affairs/cwm/ provides 24/7 confidential support, and the sexual assault resource center webpage http://sarc.usc.edu describes reporting options and other resources.

Support Systems
A number of USC's schools provide support for students who need help with scholarly writing. Check with your advisor or program staff to find out more. Students whose primary language is not English should check with the *American Language Institute* http://dornsife.usc.edu/ali, which sponsors courses and workshops specifically for international graduate students. *The Office of Disability Services and Programs* http://sait.usc.edu/academicsupport/centerprograms/dsp/home_index.html provides certification for students with disabilities and helps arrange the relevant accommodations. If an officially declared emergency makes travel to campus infeasible, *USC Emergency Information* *http://emergency.usc.edu* will provide safety and other updates, including ways in which instruction will be continued by means of blackboard, teleconferencing, and other technology.