Prerequisites

Nominally the prerequisites are CS3600, Information Assurance: Introduction to Computer Security, CS3601, Theory of Formal Languages and Automata and CS3650, Design and Analysis of Algorithms. We assume the students have some familiarity with mathematical logic, deduction and proofs.

In practice some students have not had the Automata course but have been able to complete much of the work.

Course Overview

The course has two strands:

1. We read a number of papers that discuss various security policies and “formal models” of security. We discuss the consequences of these models, their application to the design of systems and the application of the models in the production of high assurance systems.

2. We use PVS (the Prototype Verification System) from SRI to specify various security models and to prove that the models satisfy those properties. Hopefully You will see a connection between some of the papers we read and the models we study.

*gwdinolt@nps.edu*
We read all the papers listed in the references in lieu of a text along with Chapter 4 of Dorothy Denning’s book *Cryptography and Data Security* [6]. We may read additional (or replacement papers) as time and interests dictate.

I typically give a mid-term and final exam. The final is usually a “take home”, open book open notes exam. There will be several lab reports as well, see below.

**Typical Course Plan**

Actually the weeks aren’t quite right since we also will include discussions of the labs as we go along.

Week 1: Course overview, Introduction to mathematical models using a simple “MLS Router”.

Week 2: Introducing the Access Matrix Model: Read Lampson’s *Protection* paper [10] and introduce a mathematical model for the access matrix and how it might be modified

Week 3: Lattices and Information Flow: Introduce Lattices, Read and analyze D. Denning’s paper on *A Lattice Model of Secure Information Flow* [5], discuss the consequences of the theory

Week 4: Safe Systems: Discuss the Harrison, Ruzzo and Ulman paper on *Protection in Operating Systems* [9], discuss the consequences of the results


Week 6: Non-Interference: Another way of modeling security, we read Goguen/Meseguer [8], Contrast with Bell/LaPadula [1]


Week 8: Refinement in Non-Interference Models, introducing doubly labeled transition systems from the work of D. Bibighaus [2]

Week 9: Integrating the policy, the mathematical models and the specifications into the development processes.
Labs

Labs will be assigned over the course of the term. We have used the following labs in the past, but these are subject to change as time, energy and interest permit.

Lab 1: Intro to PVS and emacs, the interface to PVS, making sure you have a working installation, can execute some basic PVS command and can hand in output. PVS does not run on Windows so this may be a trial for some.

Lab 2: Proving (and finding counter examples) of some simple logical propositions (using PVS).

Lab 3: The foundation (trivial state transition model) of several of the following labs

Lab 4: A simplified version of the Access Matrix model in PVS

Lab 5: Specification Layering using our state machine model

Lab 6: The High Water Specification and proving (in PVS) a more complex sample theorem, reading and figuring out what a spec is doing

Lab 7: A more complex layering example that illustrates how concrete implementations of “state” at a lower level get used at the abstract level.

Final Exam Lab: The Chinese Wall security model, mapping the model to the spec, understanding the paper.

References


