Lecture notes for 5.5 OS assurance

Def: Assurance
- Ways of convincing others that a model, design, & implementation are correct

Methods of assurance
- Validation
- Testing
- Penetration testing
- Formal verification

Validation
- Checking that developers have implemented all requirements
- Requirements checking, design & code reviews, system testing

Testing
- Demonstrate existence of problem
- Cannot demonstrate absence of problem
- Regression testing: ensure that alterations do not break existing functionality / performance
  - regression: “going backwards”
- Heisenberg: Program modification to expose internal state for testing alters behavior
- Difficulty: test case generation
- Difficulty: code coverage
- Difficulty: exponential number of different executions
- Difficulty: different execution environments

Penetration testing
- Ethical hackers attempt to defeat security measures
- Use study, research to find problems
- Cannot demonstrate absence of problem

Formal verification

- Checking a mathematical specification of program to ensure that security assertions hold
  - Model checking, automated theorem proving
- State variables w/ initial assignment, program specification describing how state changes, boolean predicates over state variables
- [show ppt slides]
- Prove existence of problem [in spec]
- Prove absence of problem [in spec]
- Difficulty: correctness of spec
- Difficulty: exponential time & space worst case complexity
- Model checking pioneers won the 2007 Turing Award

Government security evaluations

- U.S. Orange Book (late 1970's)
  - D < C1 < C2 < B1 < B2 < B3 < A1
    - D: no protection
    - C: discretionary protection
    - B: mandatory protection
    - A: Verified protection
  - D: no requirements
  - C1, C2, B1: security features common to commercial OSes
  - B2: Proof of security of underlying model, narrative spec of TCB
  - B3, A1: Formal design & proof of TCB
- Common Criteria (2005) international standard replaced orange book
  - Originated out of European, Canadian, and US standards
  - Idea: users specify system needs, vendors implement solution and make claims about security properties, evaluators determine whether vendors actually met claims
  - Evaluation assurance level (EAL) rates systems
    - EAL1 most basic, EAL7 most rigorous
    - EAL2: RHEL 3
    - EAL3: SuSE Linux Enterprise Server V8
    - EAL4: Windows 2000, Solaris 8, RHEL 5