Exam 2 Review

CS461/ECE422 Fall 2010

Exam guidelines

- Same as for first exam
- A single page of supplementary notes is allowed
  - 8.5x11. Both sides. Write as small as you like.
- Closed book
- No calculator or other widgets.
- Students should show work on the exam. They can use supplementary sheets of paper if they run out of room.

Exam logistics

- Exam will be given during normal lecture time in 1310 DCL
- You will be given 50 minutes to complete the exam.

Topics

- Access Control
- Data base security
- Confidentiality and Integrity Policies and Models
- System Evaluation Frameworks
- Trusted System Development
- Malicious Code
- Network Security Threats and Controls

Access control

- Access Control Matrix
  - Common model for encoding protection state of system
- Access Control Lists
  - ACM by column
  - Unix and windows examples
- Capabilities
  - ACM by row
- A little bit on hardware rings. A Hardware implementation of something like the BLP and strict Biba models

Database Security

- Access control model – Griffiths and Wade model
  - Basic relational model
  - No single owner of all data/privilege
  - Use “grant” to delegate privileges
  - Use view to shared restricted set of data
  - Revocation issues
- Integrity
  - Transactions
  - Two phase commit

Trusted Models and Policies

- Mandatory Access Control
  - How does it differ from DAC
- Bell-LaPadula
  - MLS – Confidentiality policy
  - Lattice of Security Labels, e.g., Security:{Proj1, Proj2}
  - Read down, write up
  - Basic Security Theorem
  - Tranquility

Integrity Policies

- Biba models
  - Low water mark – tries to preserve indirect information flow constraints
  - Ring policy – Like low water mark but doesn't attempt to address indirect flows
  - Strict – Dual of the BLP model
- Clark Wilson
  - Design guidelines for creating high integrity systems
- Lipner Matrix model
  - Class exercise. Useful as an example of how one designs systems to use both integrity and security labels.

Evaluation Framework

- Covered TCSEC (Orange book) and Common Criteria
- Assurance vs functionality requirements
- TCSEC
  - Fixed assurance and functionality evaluation levels
- Common Criteria
  - Dynamic functionality profiles and fixed assurance levels

Design Principles

- Salzer and Schoeder's principles
  - Understand and recognize application in systems

Assurance

- Assurance is evidence that system meets requirements
- Techniques for gathering evidence during product life cycle
  - Different types of assurance: policy, design, implementation, operational
- Different development processes and how they gather assurance

Secure Software Design

- Security architecture as focus for tracking and analyzing system security
  - Security requirements
- Documentation and requirements tracing
- Threat analysis
  - Analyze design/code – identify entry points. Develop data flow diagrams
  - Identify threats
  - Build attack trees
- Security testing

Malicious Code

- Types of malicious code
  - Trojan programs
  - Rootkits
  - Virus
    - Detection and virus evasion
  - Worms
    - Propagation techniques
  - NetBots

Common Implementation Flaws

- Buffer Overflow
  - Stack smashing
- Incomplete Parameter Validation
- Time of use to time of check

- Covered a little bit on ethical hacking and vulnerability research

Network Security Concerns

- Review the network stack
- Physical/Data link layer and CIA
- Network Layer
  - Routing
  - ARP
  - ICMP
  - Smurf

Network Security Concerns

- Transport (UDP/TCP)
  - Syn flood
  - Port scan
  - DHCP
- Application
  - Spoofing
  - DNS
    - Open relay
      - Preferred server layout
    - Cache poisoning

Network Security Architecture

- Segmentation
- Perimeters and domains
- VPNs
- Common network layout
  - In, out, DMZ

Network Security Controls

- Firewalls
  - Application proxy
  - Packet filter
  - Stateful packet filter
  - NAT
- Intrusion Detection
  - Did not cover Honey pots
  - Mis-use/signature detection
  - Anomaly/statistical detection
  - IDS vs IPS

Good luck!