

Information Assurance: Review Topics for Final

Test Issues

You can use a calculator, and you can bring a single sheet of notes. It is closed book otherwise.

Look back at the review notes for exams 1 and 2 for topics through the first two tests. The final will cover all topics from the semester, although it will be slightly more biased to topics in the last third of the course.

Design Principles

Reading: Chapter 13 CS

Concepts:

- Eight security design principles proposed by Salzer and Schroeder. One set of guidelines. Another set of security design principles are the Generally Accepted System Security Principles (GASSP). You should be familiar with the 8 principles and understand how a particular design employs these principles.
 - Principle of Least Privilege
 - Principle of Fail-Safe Defaults
 - Principle of Economy of Mechanism
 - Principle of Complete Mediation
 - Principle of Open Design
 - Principle of Separation of Privilege
 - Principle of Least Common Mechanism
 - Principle of Psychological Acceptability

Introduction to Assurance

Reading: Chapter 18 CS

Concepts:

- Assurance is separate from functionality or mechanism. Assurance is the confidence that an entity meets its requirements. Functionality is how the system meets the requirements
- A trustworthy system has credible evidence that it correctly meets requirements.
- Trust is a measure of trustworthiness of a system relying on the evidence
 - Requirements must be identified
 - To argue of security assurance must create an evidence trail that proves system meets security requirements.
 - Evidence can come from a variety of sources

- Design documentation
 - Test suite results
 - Formal verification
- Assurance appears multiple places in system life cycle
 - Design, Implementation, and Operation
 - Assurance at each point in the life cycle should feed into the next stage.
- Some system life cycle models more easily gather data for high-assurance than others
 - Waterfall Life Cycle Model
 - Standard assurance document trails very easily map to waterfall
 - High assurance evidence can be gathered from other life cycle models but some additional steps may need to be added or evidence gather after the fact.

Secure System Design and Development

Reading: CS Chapter 19

Concepts:

- Best Practice Guidelines like GASSP and Security at a Glance Checklists
 - Some tools automate these checks.
- System Security Engineering Capability Maturity Model
 - Evaluate organization on good security process
 - Based on the System Engineering Capability Maturity Model
- Analyze threats to guide security requirements
 - May need to perform cost benefits analysis to determine which threats to address
 - This analysis is performed at the start of the design cycle
 - Can use requirements tracking between levels to ensure security requirements are addressed
- High level security architecture can be used to analyze integrity of security design and guide testing and reviews
 - Reference Monitor and Reference Validation Mechanism (RVM)
 - Security Kernel and Trusted Computing Base (TCB)
- Threat Modeling – similar to risk analysis
 - Analyze Entry points – associated trust level with each entry point
 - Understand Assets
 - Use Scenarios – Describe how system is expected to be used
 - Data Flow Diagrams
 - Model entry points, significant portions of system, protection domains, and how they interact
 - Threat Profiling
 - Threat tree
 - Can be used to guide which vulnerabilities have a higher payoff on mitigation
 - Can be used to guide testing
- Retrofitting security
 - Wrappers, Isolation, Interposition
- Security Testing

- Unit Testing, Functional/black box testing, Code-based or white box testing
- Security testing extra difficulties
 - often concerned with proving a negative result
 - Must think at multiple levels of abstraction
 - Vague requirements
 - Risk-based testing, uses threat trees/attack trees to guide testing
 - Test mitigations for highest risk threats first
- Test Coverage
 - Covering error cases is particularly hard

System Evaluation

Reading: CS Chapter 21

Concepts:

- Many computer security products pass through a formal evaluation. It is important to understand what evaluation does and does not promise. The TCSEC evaluation provides historical background. Common Criteria is the evaluation scheme used today.
- System evaluation is for the benefit of the end user of the system
 - Rely on vendor/developer evidence
 - Rely on third party expert
 - Informal review
 - Formal evaluation
- Formal evaluation – provides systematic framework for evaluation and comparison
 - TCSEC or Orange Book
 - Relies on following mechanisms
 - Bell-LaPadula
 - Reference Monitor
 - Trusted Computing Base
 - Functional Requirements
 - DAC
 - Object Reuse
 - MAC and Labels
 - Identification and Authentication
 - Audit
 - Trusted Path
 - Assurance Requirements
 - Configuration Management
 - Trusted Distribution
 - System Architecture
 - Design Specification
 - Verification
 - Testing
 - Product Documentation
 - Fixed set of classes defined
 - D through A1

- Each class has a set of function and assurance requirements defined
- Common Criteria
 - Internationally recognized
 - Functional and assurance requirements similar to TCSEC
 - Functionality classes are not fixed
 - Defined per product in a security target
 - Security target can be based on a protection profile
 - Which is also evaluated
 - Fixed set of assurance levels: EAL1 through 7
 - Product or protection profile is evaluated with respect to a EAL

Code Vulnerabilities and Malicious Code

Reading: CS Chapter 22, part of Chapter 19, and part of Chapter 23.

Concepts:

- Various types of malicious codes
 - Trojans
 - Key loggers
 - Rootkits
 - Virus – propagates by infecting other files
 - Dormant, propagation, triggering, and execution phases
 - Signature scanning for protection
 - Polymorphic or stealth virus
 - Worms – propagate directly from one computer to another
 - Identifying potential victims
 - Randomly scan addresses
 - Not practical in larger address space of IPv6
 - Hit list scanning
 - DNS searches or spiders
- Configuration Management – Source control mechanisms or processes
- Malicious vs non-malicious program errors
- RIOS Taxonomy of program errors
 - Incomplete parameter validation
 - Inconsistent parameter validation
 - Implicit sharing of privileged/confidential data
 - Asynchronous validation/inadequate serialization
 - Inadequate identification/authentication/authorization
 - Violable prohibiting/limit
 - Exploiting logic error
- How buffer overflows can be exploited to inject malicious code
 - Specifically how can stack overflows be exploited
 - Buffer overflow protections
 - Write correct code
 - Use appropriate language
 - Tools like Libsafe

- Rely on hardware
 - Address Space Randomization
- Time of Check to Time of Use (TOCTOU) errors
- Vulnerability Research and Ethical Hacking
 - Software Fault Injection and exploit research frameworks

Auditing

Reading: CS Chapter 24 or Intro Chapter 21

Concepts:

- Log is the data collected during the operation of a system
- Audit is the analysis performed on the log to ensure that system security policies are met
- Audit system components
 - Logger – Generates the log
 - Analyzer – Analyzes the logs
 - Notifier – Alerts based on results of the analyzer
- Use Security Policy to guide what needs to be logged
 - E.g., Bell-LaPadula rules
 - Create action => condition rules to determine what needs to be logged
- Log Sanitation – two cases
 - Sensitive data cannot leave site
 - Sensitive data cannot leave system
 - Anonymizing sanitizer – Cannot recover data
 - Pseudonymizing sanitizer – Can recover data
- Adding in security auditing later
 - Not necessarily clear what the security policy should be
 - Can use known attacks to guide what needs to be logged.
- State-based auditing vs transition-based auditing
- Audit browsing

Physical Security

Reading: Secrets of Computer Espionage Chapters 5, 12, and 13; Steganography tutorial; Soft Tempest article.

Concepts:

- Computer systems interact with the physical world. Without considering physical security, strong computer security is not very effective.
- Consider physical access
- Hiding information on a file system
 - Use unusual extensions
 - Root kit techniques
 - Encryption

- Steganography – Hide information in other data sources
 - Digital Watermarking uses a similar mechanism
 - Deleting files and scrubbing disks
 - Backup issues
- Paper disposal requirements
- Copier/printer/fax security issues
 - Physical access
 - Imprints on ribbons
 - Labeled output devices
- Phone Security
 - IP phones
 - Physical access to handsets
 - Cell phones
- Emanations security EMCSEC or Tempest
 - Emanations from computer monitors
 - Van Eck
 - Select specific pixel values to produce AM frequencies
 - Embedding data in dither
 - Protection
 - Shielding
 - Anti-tempest fonts to reduce high frequencies from the font
 - Physical separation of sensitive and less sensitive equipment

Privacy

- The right to be left alone
- Anonymity technology
 - MIX networks – provide unlinkability. Attacker does not know which pairs are communicating
 - MIX cascade – Concern about attacker controlling some mixers
 - Dining Cryptographers – Anonymous publishing
 - Randomized Routing - unlinkability
 - Onion Routing
 - Crowds, Tor
- Digital cash – a user of anonymity technology

Hardware-Enforced Security

Reading: Pentium II Software Developer's manual, TCG Specification architecture

Concepts:

- Judicious use of hardware can greatly enhance the security of a system
- Hardware-enforced security in General Purpose processors
 - Protection rings/privilege levels in Pentium architecture
 - Associate levels with various memory segments

- Enforce access rules in hardware
 - Read-only /read-write page bits
 - No Execute bits
 - Hardware support for capabilities (access control)
- Physically separate security processor
 - Store keys and potentially perform crypto operations
 - E.g., boot-strap trust for encrypted files
 - Physically tamper-proof
 - Fortezza crypto card or smart card
 - Portable, used for authentication
 - Secure Co-processor
 - Statically located on mother board
- Trusted Computer Group propose standards for OS to use secure co-processor
 - Co-processor called the Trusted Platform Module (TPM)
 - TPM includes Endorsement Key and can generate Attestation Identity Key
 - Associated with particular TPM instance
 - Transitive trust
 - Bootstrap integrity measurements from TPM to whole system
 - TPM message operations
 - Binding, Signing, Sealing, and Sealed-signing
 - TPM proposed use for Digital Rights Management