   f. Type checking
7. Computerized Vermin
   a. Trojan horse
   b. Computer virus
   c. Computer worm
   d. Bacteria
   e. Logic bomb
8. UNIX Practises
   a. Programming environment: PATH, IFS
   b. Checking software for potential problems
9. Trust
10. Network Security
   a. Privacy Enhanced Electronic Mail, PGP
   b. Public key management, including certificates, the binding of a name to a principal (user), and certificate management schemes
   c. Digital signatures (what it is)
11. Security in Programming
   a. Unknown interaction with other system components
   b. Overflow (both numeric and buffer)
   c. Race conditions (TOCTTOU flaw)
   d. Environment (shell variables, UIDs, file descriptors, etc.)
   e. Not resetting privileges
12. Vulnerabilities Models
   a. RISOS
   b. PA
   c. Uses
13. Penetration Studies
   a. Relationship to formal verification and testing
   b. Flaw Hypothesis Methodology
   c. Using vulnerabilities models
14. Intrusion Detection Systems
   a. Anomaly detection
   b. Misuse detection
   c. Specification detection

Study Guide for Final

This is simply a guide of topics that I consider fair game for the final. I don't promise to ask you about them all, or about any of these in particular; but I may very well ask you about any of these.

1. Fundamentals
    a. Basics of risk analysis
    b. Saltzer and Schroeder's design principles
    c. Relationship of security policy to security
    d. Basics of robust programming
2. Cryptography
    a. Types of attacks: ciphertext only, known plaintext, chosen plaintext, chosen ciphertext
    b. Types of ciphers: substitution, transposition, product (both substitution and transposition)
    c. Goal of ciphers; what makes a cipher theoretically unbreakable
    d. Caesar cipher, Vigenère cipher, one-time pad
    e. What the DES is, characteristics
    f. Public key cryptosystems
    g. RSA
    h. Confidentiality and authentication with secret key and public key systems
3. User and System Authentication
    a. One-way hash functions (cryptographic hash functions)
    b. UNIX password scheme, what the salt is and its role
    c. Password selection, aging
    d. Challenge-response schemes
    e. Attacking authentication systems: guessing passwords, spoofing system, countermeasures
    f. UNIX real, effective, saved, audit UIDs
4. Privileges
    a. Setuid, setgid
    a. Role accounts
    a. Nesting programm units
5. Access Control
    a. Multiple levels of privilege
    b. UNIX protection scheme
    c. ACLs, capabilities, lock-and-key
    d. MULTICS ring protection scheme
    e. MAC, multilevel (military) security model; lattices
    f. Differences between MAC, DAC, ORCON
    g. Bell-LaPadula model
6. Integrity Models
    a. Biba's model
    b. Clark-Wilson model
    c. Chinese Wall model
    d. File signature generation (integrity checksumming, etc.) and checking
    e. Safe practises ("safe hex")