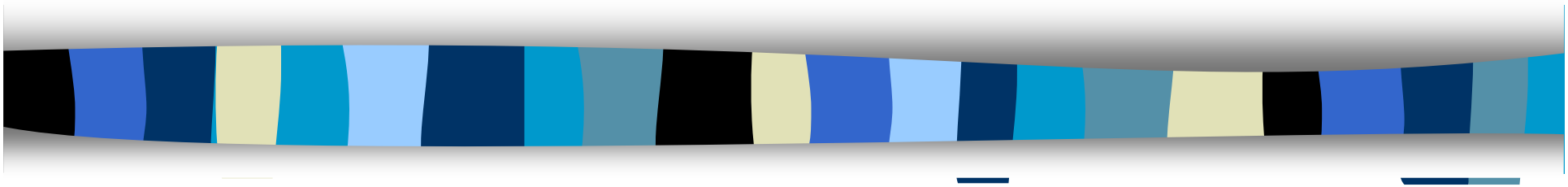
Computer Security
CS 426



Review for Final Exam

Basic Concepts

- Confidentiality
- Integrity
- Availability

- Authenticity ≈ Integrity (in communications)
- Non-repudiation
- Privacy (general concept, need to defined for different contexts)
  - K-Anonymity ?

Cryptography Basics

- Basic terminology: plaintext, ciphertext, ciphers
- Adversarial models:
  - Ciphertext-only, known-plaintext, chosen plaintext, chosen ciphertext
- Brute-force attacks
- General Mono-alphabetic Substitution Cipher
  - Breakable by frequency analysis in ciphertext-only attacks
- Information theoretic security
  - Known as perfect secrecy for ciphers
    - Requires key length >= msg length (proof idea)

OTP & Stream Ciphers

- One-Time Pad (OTP)
  - Provides perfect secrecy
  - Two-time pad is insecure
- Stream cipher
  - Simulate OTP by using PRNG
  - Weaknesses: malleable, need to be very careful to avoid becoming two-time pad
- PRNG: should satisfy the next-bit test.

Block Ciphers

- Block ciphers
  - Use a larger block size to defeat frequency analysis
  - Aim at providing Pseudo Random Permutation (PRP)
- DES: 56-bit key size, 64-bit block size, considered insecure now b/c bruteforce attack
- Brute-force: exhaustive key search, dictionary attack
- AES: block size, key sizes, no known weaknesses
- Encryption modes: ECB, CBC, CTR
  - How they work?
  - ECB insecure because of deterministic encryption
  - CBC and CTR randomized, secure, CTR builds stream cipher from block cipher

Cryptographic Hash Functions

- Apply to arbitrary long messages, output fix-length digests.
- Three properties (what they mean, how much time to break them in brute-force fashion):
  - preimage resistant (one-way)
  - 2-nd preimage resistant (weak collision resistant):
  - collision resistant (strong collision resistant):
- Well know hash functions: MD5 (128 bits, collision resistance broken), SHA-1 (160 bits, collision resistance appear soon to be broken), SHA2 family
- Message authentication code
  - Why needed? Security requirements?
  - HMAC construction

Public Key Cryptography

- PK Encryption: two keys, can check whether two keys are a pair, but cannot compute private key from pubic key
- How RSA works: pub key: (n=pq,e), pri key (d)
- RSA security: depends on factoring, how long should the modulus n=pq be.
- RSA security: direct usage violates IND, use OAEP (how it works)
- Usage of RSA & secret-key encryption
- Diffie-Hellman key agreement (subject to active attacks)
- How El Gamal encryption works?

Digital Signatures & Key Distributions

- Non-repudiation (why MAC doesn't work)
- How RSA signatures work?
- How to sign hashes of messages?
- Why want the hash function to be collision resistant?
- In key distribution, what does the TTP need to do in the symmetric key setting?
  - Scalability of the approach.
- Public key distribution
  - Weaknesses of directory & public announcements
  - Usage of public key certificates. How they work?
  - Nature of trust in Root CA. Compare with symmetric case.

Crypto Topics Not Required in Final Exam

- Quantum cryptography
- Details of Needham-Schroeder and Kerberos
- Commitment & Zero-knowledge Proofs

Operating System Security

- Goal 1: multiple users sharing one computer
- Goal 2: secure operation in networked environment
- CPU modes: kernel mode vs. user mode
- System calls
- Privileged processes vs. privileged code

User Authentication

- Types of authentication: know, have, are
- Threats to passwords
  - Online guessing, offline dictionary, spoofing, shoulder surfing, social engineering
- UNIX storage of passwords
  - Addition of /etc/shadow in addition to /etc/passwd
  - Usage of salts
- Other defenses
  - Disabling account after multiple failed attempts
  - Mechanisms to avoid weak passwords
- Trusted path
- Lamport's One-time Passwords scheme

UNIX Access Control

- Reference monitor properties:
  - Complete mediation (non-bypassable)
  - Tamper-proof
  - Small enough to be analyzable
- Users, principals, and subjects
- Permission bits: r/w/x on files and directories, suid on files
- Processes: how euid changes
- Need for setuid programs, violating least-privilege, potential vulnerabilities

Software Vulnerabilities

- Buffer overflow
  - Overflow ret address to shell code, return to libc, off by one, function pointers, heap overflow
- Effectiveness of different defenses
  - Type safe languages, safe library functions, non-executable stack, StackGuard (using canary), Address space layout randomization, Instruction set randomization
- Integer overflow

- No question on types of malwares and their details.

Market Failure of Software Security

- What is market failure?
- Why incentives are not aligned to make secure hardware?
  - No measurement of security, needs to ship product early, no liability, patching costs little, testing/debugging expensive, buggy software forces users to ungrade
- What would help?

Access Control

- Discretionary access control vs. mandatory access control
- DAC is vulnerable to trojan horses and buggy programs
- BLP model: simple security condition, *-property, trusted subjects, covert channels
- Security lattices

Trusted Systems

- Trusted computing base
- TCSEC: D, C1, C2, B1, B2, B3, A1
  - B requires MAC, A requires formal verification (no other details required)
  - Limited to OS
  - Combine functionality and assurance in a single linear scale
- Common Criteria
  - TOE, PP, ST, EAL
  - Windows EAL4+

Biba, Clark-Wilson, and Chinese Wall

- Integrity levels different from security levels in BLP
- Five policies in Biba (what they are)
- Two meanings of integrity levels on objects
- Difference between confidentiality and integrity
  - Integrity has to trust subjects
- Clark & Wilson
  - Two high-level mechanisms: Well-formed transactions and separation of duty
  - Concepts: UDI, CDI, IVP, TP
- Chinese wall: avoid COI

SELinux, UMIP, and IFEDAC

- Type enforcement idea: use binary to determine access
- What is an assured pipeline policy
- UMIP: security objective (network-based attacks)
  - Basic UMIP policy.
  - How integrity of subjects & objects changes?
- Gap between requests and policies
- Why UNIX DAC's approach of maintaining one responsible principal is problematic.
  - Implicit assumptions of DAC: software are benign and correct
- IFEDAC: integrity level tracking and access rules
  - Comparison with Biba

RBAC

- RBAC96 models
  - Features in the four models
  - How to determine session permissions, and which roles can be activated.
  - Semantics of role hierarchies
  - Mutual exclusion constraints

- RBAC manages how to assign permissions to principals, unlike DAC and MAC, which primarily focus on how to authorize subjects

Network Security

- ARP spoofing
- TCP sequence prediction attacks
  - Session hijacking
- DOS attacks
  - SYN flooding, Smurf, reflection, DDoS, pulsing

DNS Attacks

- Fundamental reason:
  - Integrity/authenticity of DNS response messages not guaranteed
- Attack 1: control one evil domain, return malicious answers when queried, defended by the Bailiwicks Rules
- Response racing attacks
  - Need to guess transaction IDs
  - Attack based on the birthday paradox
  - Kaminsky attack (why effective, why different from previous)
- Defense: source port randomization. Benefits and limitations.
- Defense: 0x20 encoding.

Network Defenses & IDS

- Firewalls/ IDS/ IPS
- Terminologies: DMZ, VPN, and why need them.
- Why need defense in depth, beyond perimeter defense.
- Stateless/stateful/proxy/personal firewalls
- IDS assumptions:
  - System activities are observable
  - Normal and intrusive activities have distinct evidence
- Misuse (signature based) vs. Anomoly
- NIDS vs. HIDS
- Detection rate and false positive rate.
- Exploit-based sig vs. vulnerability-based sig

Web Security

- SQL injection
  - How it works?
  - How to avoid such attacks when writing new apps?
  - How does the CANDID approach work?
- Cross Site Scripting
  - If we reach the topic in Friday