Nathan Tung

CS111 Chapter 11 Overview, Section 11.2 Outline #11

Chapter 11.2 (11-28 – 11-36), 11.4 (11-49 – 11-54), 11.6 (11-72 – 11-80)

- Authenticating Principles
    - Guard asks questions
        - Who is the principal making the request? Is it actually from that person?
    - Separating trust from authenticating principles
    - Authenticating principals
        - Check real-world identity of principal
            - Physical property (voice recognition); something one has (magnetic cards), knows (PIN)
        - Use verification of identity
    - Cryptographic hash function can protect passwords
        - Secure Hash Algorithm (SHA)
        - Computationally secure (infeasible to break into)
        - Work factor: quantifier used to measure computationally security
    - Design principle: minimize secrets
    - Dictionary attack used by brute-forcers (effective, since most passwords suck)
    - Protocol between user and service
        - Authenticates principal to guard
        - Authenticates service to principal
        - Password never travels the network
        - Password only used once per session
- Message Confidentiality
    - Encryption: plaintext → ciphertext
    - Decryption: ciphertext →plaintext
    - ENCRYPT, DECRYPT are implemented using cryptographic transformations
        - Observers cannot translate the messages, even if intercepted
    - Public key and private key
        - Bob hands over public key to Alice, who uses it to encode her message
        - Only Bob has his own private key to decode a message encoded via public key
    - Must be able to withstands attacks like…
        - Ciphertext-only attack: adversary has ciphertext and encrypt/decrypt algorithms; he should not be able to use redundancies and patterns to reconstruct the plaintext
        - Known-plaintext attack: adversary has ciphertext and some of the plaintext; he should not be able to determine the key
        - Chosen-plaintext attack: adversary has ciphertext and chosen corresponding plaintext; he may convince someone to send an encrypted message of plaintext to his liking (adaptive attack)
            - Oracles, which encrypts inputs without question, might encourage adversary attacks
        - Chosen-ciphertext attack: adversary has ciphertext and observes plaintext when it's converted
    - Confidentiality and Authentication
        - Confidentiality: Alice encrypts message
            - Unlikely, confidential info should have a guarantee of where it's from
        - Authentication: Alice signs message
            - Common, since a lot of data is public, but its origin is needed or helpful
        - For both, we must encrypt and sign the message (in this order, to cover up weaknesses)
            - Use different keys for authentication/encryption
            - Explicitness principle: specify who the message is from/for
- Authorization: Controlled Sharing
    - Separate programs that share sensitive data?
        - This restriction on sharing is impossible
        - Rather, use trusted and untrusted networks
        - Still not very convenient
    - Authorization Operations
        - Authorization: gains principal permission to operate on object

        - Mediation: operation checks whether principal has operation permission on object
        - Revocation: decision that removes previous permission from principal
- Simple Guard Model
    - Authorization matrix with principal rows and object columns
    - Discretionary access-control systems: object creator assigns permissions for that object
    - Non-discretionary: another chosen authority has that responsibility
    - Two instances
        - List systems: organized by column
            - Revocation is less disruptive than ticket system
            - Guard holts list of tokens corresponding to authorized principals
            - Tokens list (access-control list) dictates who has access
            - Requires searching a list, but revocation is easy
        - Ticket systems: organized by row
            - Guard holds ticket (capability) for its object, which it can give out/take back
            - Tickets can get passed around (good and bad), but revocation is hard
        - Tokens and tickets must be shielded from forgery
        - Agency allows a switch between list and ticket systems
        - Protection groups: principals used by more than one user
            - Might be implemented using an ACL
            - Kind of like a permission group
    - UNIX principals identified by strings or integer names (UIDs)
    - GID is the principal for whom the process is running
    - Use principal of least privilege
    - Authentication users based on UID, GID, and password
    - Access Control Check
        - Superuser (UID 0) has permissions by default
        - Matching UID of owner and process prompts for ACL check
        - Matching GID of process and file prompts for ACL check
        - UID/GID don't match, so we check ACL for other matches
- Caretaker Model
- Flow-Control Model