Lecture 23 Outline

May 23, 2018

Reading: §12, 16

Assignments: Homework 4, due on May 25, 2018 at 11:59pm
Lab 3, due on May 23, 2018 at 11:59pm

1. Attacks
   a. Speeding up guessing: rainbow tables
   b. Scavenging: passwords often typed where they might be recorded as login name, in other contexts, etc.
   c. Ask the user: very common with some public access services
2. Defenses
   a. For trial and error at login: dropping or back-off
   b. For thwarting dictionary attacks: salting
3. Password aging
   a. Pick age so when password is guessed, it's no longer valid
   b. Implementation: track previous passwords vs. upper, lower time bounds
4. Ultimate in aging: One-Time Password
   a. Password is valid for only one use
   b. May work from list, or new password may be generated from old by a function
5. Challenge-response systems
   a. Computer issues challenge, user presents response to verify secret information known/item possessed
   b. Example operations: $f(x) = x + 1$, random, string (for users without computers), time of day, computer sends $E(x)$, you answer $E(D(E(x)) + 1)$
   c. Note: password never sent over network
6. Biometrics
   a. Depend on physical characteristics
   b. Examples: pattern of typing (remarkably effective), retinal scans, etc.
7. Location
   a. Bind user to some location detection device (human, GPS)
   b. Authenticate by location of the device
8. Access Control Lists
   a. UNIX method
   b. Full ACLs: describe, revocation issue
9. Capabilities
   a. Capability-based addressing
   b. Inheritance of C-Lists
   c. Revocation: use of a global descriptor table
10. Lock and Key
    a. Associate with each object a lock; associate with each process that has access to object a key (it's a cross between ACLs and C-Lists)
    b. Example: cryptographic (Gifford). X object enciphered with key K. Associate an opener R with X. Then:
    OR-Access: K can be recovered with any D_i in a list of n deciphering transformations, so $R = (E_1(K), E_2(K), \ldots, E_n(K))$ and any process with access to any of the D_i's can access the file
    AND-Access: need all n deciphering functions to get K: $R = E_1(E_2(\ldots En(K)\ldots))$
    c. Types and locks
11. Secret sharing