Lecture 24 Outline

Reading: *text*, §12, 15 **Assignments due:** Homework 4, due May 23, 2011

1. Attacks
   a. Exhaustive search: password is 1 to 8 chars, say 96 possibles; it's about 71016
   b. Inspired guessing: think of what people would like (see above)
   c. Random guessing: can't defend against it; bad login messages aid it
   d. Scavenging: passwords often typed where they might be recorded as login name, in other contexts, etc.
   e. Ask the user: very common with some public access services
2. 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   a. Pick age so when password is guessed, it's no longer valid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
3. 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
4. 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 on wire or network
5. Biometrics
   a. Depend on physical characteristics
   b. Examples: pattern of typing (remarkably effective), retinal scans, etc.
6. Location
   a. Bind user to some location detection device (human, GPS)
   b. Authenticate by location of the device
7. Access Control Lists
   a. UNIX method
   b. ACLs: describe, revocation issue
8. Capabilities
   a. Capability-based addressing
   b. Capability-based addressing
   c. Inheritance of C-Lists
   d. Revocation: use of a global descriptor table
9. 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: use crypto (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
      $R = (E_1(K), E_2(K), \dots, E_n(K))$ 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: $R = E_1(E_2(\dots E_n(K)\dots))$
   c. Types and locks