Outline for May 20, 2004

Reading: Chapters 12, 14

Discussion Problem

One of the concerns in electronic voting is the integrity of the vote count. Most electronic voting schemes work by having a computerized voting machine take a user's votes through an input device like a touch screen and record it in storage. All votes are stored on three different sets of media, for redundency. At the end of the day, one of the pieces of media is removed from the machine, and the votes on it are uploaded to a server using a modem and telephone line. The server is not on the Internet, and can only be accessed through the phone line when the operators are told to turn the modem at the server's end on.

1. One contentious issue is the need for a voter-verifiable audit trail, which is a record of how the voter voted in a form that the voter can read or hear. Is this really necessary?
2. If you were an analyst and asked to check the integrity of the electronic voting system, where would you look for potential flaws?

Outline for the Day

1. Passwords
   a. How UNIX does selection
   b. Problem: common passwords
   c. May be pass phrases: goal is to make search space as large as possible, distribution as uniform as possible
   d. Other ways to force good password selection: random, pronounceable, computer-aided selection
   e. Go through problems, approaches to each, *esp*. proactive
2. Password Storage
   a. In the clear; MULTICS story
   b. Enciphered; key must be kept available; get to it and it's all over
   c. Hashed; present idea of one-way functions using identity and sum; show UNIX version, including salt
3. Attack Schemes Directed to the Passwords
   a. Exhaustive search: UNIX is 1-8 chars, say 96 possibles; it's about 7e16
   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
4. 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   a. Pick age so when password is guessed, it's no longer valid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
5. 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
   c. Example: S/Key
6. 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
   d. Attack: monkey-in-the-middle
   e. Defense: mutual authentication

7. Biometrics
   a. Depend on physical characteristics
   b. Examples: pattern of typing (remarkably effective), retinal scans, *etc*.
8. Location
   a. Bind user to some location detection device (human, GPS)
   b. Authenticate by location of the device