Lecture Notes 2 : Lecture of 14 September 1999

September 21, 1999

1 User Identification and Authentication (passwords and biometrics)

- Who is user? (accountability, authorization)
- Which system? (mutual identification and authentication)
- Can user prove who he is – that is, authenticate himself?

1.1 Issues

- Error rates? (false positives and false negatives)
- How does authentication persist across the session? Can adversary "hijack" session after authentication?
- What security requirements are there for system? (authentication database)

1.2 Classic Bases for Authentication

- something you know (password, PIN)
- something you have (token, smart card, key)
- something you are (biometrics)
- where you are (physical location, specific terminal)

Examples:

- ATM
- PIN (something you know)
- card (something you have)
- iris scan (something you are)

- location (physical)
- audit, e.g., video record

Calling cards are an example of combined ID and Password. They represent a "modest" fence...which is getting more modest by the day!

1.3 Problems with Passwords

- "guessing" of password by attacker
- compromise of system's password file
- theft, accidental disclosure, forced disclosure
- absence of continuing authorization (vulnerable to session hijacking)
- "hard to guess" = "hard to remember"

1.4 Hashing

Hashing is a technique in which a "hash" of the actual password, instead of the password itself, is stored in the system file. The hash function is typically a public, one-way function. That is, knowing the password x, one can readily determine the hash of that password, h(x). However, the converse is NOT true. Knowing h(x), one cannot easily determine x.

Ideally, it is also difficult to "guess" x' such that h(x') = h(x). Hashes typically map to large spaces (e.g., 128 bits).

1.5 Types of Attacks

- online: an online attack requires active interaction with the target. Examples include connecting to the login port and attempting a login, or snooping on the connection between two target computers.
- offline: an offline attack is one in which the attacker can proceed without any interaction with the target. For instance, the attacker can obtain the password file of the target machine, and then spend a week trying to brute force one of the password entries on his local computer. The target machine has no way of knowing that this computation is happening.

2 Defenses Against Online Attacks

- limit the number of bad password attempts allowed. (This may result in vulnerability to denial of service attacks.)
- limit rate at which attempts can be made to log in.

- use system-generated passwords (can be hard to remember)
- enforce periodic password changes (can cause user frustration)
- use a "shadow" password file which is available only to routines that are authorized to modify the file. The public version of the password file is stripped of sensitive password data, and contains only the user's name, his shell preference, and similar information.

2.1 Attack Techniques and Possible Defenses

- One attack method is to pre-hash each word in the dictionary. After downloading the password hash file from the system, compare files, seeking a match. One defense against this sort of attack is to use a "salt," or a random value assigned to each user. The system password file would then contain: Username, salt, h(salt:user). This approach would force an attacker to check each possible hashed value for each user, encouraging the attacker to start the attack only after the password entry has been obtained.
- A second means of attack is through Theft or Disclosure of Password. Defense against this include use of encryption to protect against "snooping" on the wire, system display of last login time, system reporting of information lost, tracking of patterns of use (intrusion detection), and designation of a "panic password" to be used by legitimate users who are under duress. Other defenses against this sort of attack include use of non-static passwords. Non-static passwords can include a list of arbitrary passwords, a list of related passwords, time-varying passwords, or challenge-response arrangements. Mutual authentication is also a useful technique to guard against theft or disclosure of passwords. This is a "trusted path" problem in which the user needs to be able to confirm that the legitimate system is asking for his or her password. Approaches include "distinguished windows," in which the legitimate sign-on window is somehow identifiable (e.g., borderless), or use of a reserved sequence of keystrokes to obtain a login screen.