429 CSS-3 Computer Security

Course Instructor: Maha Alwuthaynani

Lecture 6: User Authentication
• User Authentication is the process of verifying an identity claimed by or for a system entity.

• An authentication process consists of two steps:
  • **Identification step**: Presenting an identifier to the security system. (Identifiers should be assigned carefully, because authenticated identities are the basis for other security services, such as access control service.)
  • **Verification step**: Presenting or generating authentication information that corroborates the binding between the entity and the identifier.
There are four general means of authenticating a user’s identity, which can be used alone or in combination:

- **Something the individual knows**: Examples include a password, a personal identification number (PIN), or answers to a prearranged set of questions.

- **Something the individual possesses**: Examples include electronic key cards, smart cards, and physical keys. This type of authenticator is referred to as a *token*.

- **Something the individual is (static biometrics)**: Examples include recognition by fingerprint, retina, and face.

- **Something the individual does (dynamic biometrics)**: Examples include recognition by voice pattern, handwriting characteristics, and typing rhythm.
Password-Based Authentication

• The password serves to authenticate the ID of the individual logging on to the system.

• The system compares the password to a previously stored password for that user ID.

• The ID provides security in the following ways:
  • ID determines whether the user is authorized to gain access to a system.
  • ID determines the privileges accorded to the user.
  • ID is used in access control.
Attacks against password-based authentication are the following:

- **Offline dictionary attack:** The attacker obtains the system password file and compares the password hashes against hashes of commonly used passwords. If a match is found, the attacker can gain access by that ID/password combination.

- **Specific account attack:** The attacker targets a specific account and submits password guesses until the correct password is discovered.

- **Popular password attack:** The attacker chooses a popular password and try it against a wide range of user IDs.
Password Vulnerabilities

• **Password guessing against single user:** The attacker attempts to gain knowledge about the account holder and system password policies and uses that knowledge to guess the password.

• **Exploiting user mistakes:** If the system assigns a password, then the user is more likely to write it down because it is difficult to remember. A user may intentionally share a password, to enable a colleague to share files. This situation creates the potential for an adversary to read the written password.

• **Electronic monitoring:** If a password is communicated across a network to log on to a remote system, it is vulnerable to eavesdropping.
Hashed Passwords

- A widely used password security technique is the use of hashed passwords and a salt value.
- To load a new password into the system, the user selects a password and this password is combined with a fixed-length called salt value.
- In older implementations, salt value is related to the time at which the password is assigned to the user.
- Newer implementations use a random number as a salt value.
- The password and salt serve as inputs to a hashing algorithm to produce a fixed-length hash code. The hashed password is then stored together with a plaintext copy of the salt, in the password file for the corresponding user ID.
Hashed Passwords

- When a user attempts to log on to a UNIX system, the user provides an ID and a password.
- The operating system uses the ID to index into the password file and retrieve the plaintext salt and the encrypted password. The salt and user-supplied password are used as input to the encryption routine. If the result matches the stored value the password is accepted.

**The salt serves three purposes:**
- It prevents duplicate passwords from being visible in the password file.
- It greatly increases the difficulty of offline dictionary attacks.
- It becomes nearly impossible to find out whether a person with passwords on two or more systems has used the same password on all of them.
Password Cracking

- Password cracking is the traditional approach to password guessing.
- It aims to develop a large dictionary of possible passwords and to try each of these against the password file.
- Each password must be hashed using each salt value in the password file and then compared to stored hash values.
- If no match is found, then the cracking program tries variations on all the words in its dictionary of likely passwords.
Password File Access Control

• One way to thwart a password attack is to deny the opponent access to the password file.

• If the hashed password portion of the file is accessible only by a privileged user, then the opponent cannot read it without already knowing the password of a privileged user.

• The hashed passwords are kept in a separate file from the user IDs, referred to as a shadow password file.

• Shadow password file should be protected from unauthorized access.
Objects that a user possesses for the purpose of user authentication are called tokens.

**Types of Cards Used as Tokens:**
- Memory Cards
- Smart Cards
Memory Card

- Memory cards can store but not process data.
- The most common such card is the bank card with a magnetic stripe on the back.
- A magnetic stripe can store only a simple security code, which can be read by an inexpensive card reader.
- There are also memory cards that include an internal electronic memory.
- The memory card, when combined with a personal identification number (PIN) or password, provides significantly greater security than a password alone.

The potential drawbacks are the following:
- Requires special reader
- Token loss
- User dissatisfaction
A smart card has the appearance of a credit card
It contains within it an entire microprocessor, including processor, memory, and I/O ports.
In some cards, the I/O ports are directly accessible by a compatible reader by means of exposed electrical contacts.
Other cards rely instead on an embedded antenna for wireless communication with the reader.

A typical smart card includes three types of memory:

- **Read-only memory (ROM)** stores data that does not change during the card’s life such as the card number and the cardholder’s name.
- **Electrically erasable programmable ROM (EEPROM)** holds application data and programs, such as the protocols that the card can execute. It also holds data that may vary with time.
- **Random access memory (RAM)** holds temporary data generated when applications are executed.
The typical interaction between a smart card and a reader or computer system:

- Each time the card is inserted into a reader, a reset is initiated by the reader to initialize parameters such as clock value.
- After the reset function is performed, the card responds with answer to reset (ATR) message.
- This message defines the parameters and protocols that the card can use and the functions it can perform.
- The terminal may be able to change the protocol used and other parameters via a protocol type selection (PTS) command.
- The cards PTS response confirms the protocols and parameters to be used.
- The terminal and card can now execute the protocol to perform the desired application.
A biometric authentication system attempts to authenticate an individual based on his or her unique physical characteristics.

These include:

- Static characteristics, such as fingerprints, hand geometry, facial characteristics, and retinal and iris patterns
- Dynamic characteristics, such as voiceprint and signature.
Operation of a Biometric Authentication System

(a) Enrollment

(b) Verification

(c) Identification