SLIIT
Discover Your Future

IE2022 - Introduction to Cyber Security

Lecture - 09

User Authentication

Mr. Amila Senarathne

Reading Assignment:

- W. Stallings and L. Brown, “Computer Security, Principles and Practice,, Pearson, Chapter 3.
- Other related materials

Authentication - Definition (RFC 2828)

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.

Identification

- An ID provides security because
  - The ID determines if the user is authorized to access the system
  - The ID determines the privileges given to the user

    e.g., superuser has the highest privilege while guest/anonymous has the least privilege
  - The ID is used as discretionary access control

    e.g., access rights (read, write, execute) to a file

Vulnerabilities of I&A

Some of I&A's more common vulnerabilities that may be exploited to gain unauthorized system access include:

- Weak authentication methods
- The potential for users to bypass the authentication mechanism
- The lack of confidentiality and integrity for the stored authentication information
- The lack of encryption for authentication and protection of information transmitted over a network
- The user's lack of knowledge on the risks associated with sharing authentication elements (e.g., passwords, security tokens)

Means of Authentication

There are four general means of authenticating a user’s identity, which can be used alone or in combination:

- **Something the individual knows:** Examples includes 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

Most widely used means of authentication

- The system maintains a password file indexed by ID
- Typically the system stores one-way hash function of the password
- When a user enters a password, the system compares it with the password for the ID in the file

Authentication using passwords is vulnerable to attacks

Vulnerabilities of Passwords

Offline dictionary attack

- This attack is possible if the hacker can gain access to the system's password file and compares the password hash against the hashes of common words
- Countermeasures : controls to prevent unauthorized access to the password file, intrusion detection measures to identify a compromise, and rapid reissuance of passwords should the password file be compromised.

Specific account attack:

- The attacker targets a specific account and submits password guesses until the correct password is discovered.
- Countermeasures : account lockout mechanism, which locks out access to the account after a number of failed login attempts. Typical practice is no more than five access attempts.

Vulnerabilities of Passwords

Popular password attack

- Use a popular password and try it against a wide range of user IDs. A user's tendency is to choose a password that is easily remembered
- Countermeasures: Password policies and scanning the IP addresses of authentication requests and client cookies for submission patterns.

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.
- Countermeasures: training in and enforcement of password policies that make passwords difficult to guess. Such policies address the secrecy, minimum length of the password, character set, prohibition against using well-known user identifiers, and length of time before the password must be changed.

Vulnerabilities of Passwords

Exploiting user mistakes

- User is more likely to write it down because it is difficult to remember. A user may intentionally share a password.
- Also, attackers are frequently successful in obtaining passwords by using social engineering tactics that trick the user or an account manager into revealing a password. Many computer systems are shipped with preconfigured passwords for system administrators.
- Countermeasures: user training, intrusion detection, and simpler passwords combined with another authentication mechanism.

Workstation hijacking

- The attacker waits until a logged-in workstation is unattended.
- Countermeasures : automatically logging the workstation out after a period of inactivity and Intrusion detection schemes can be used to detect changes in user behavior.

Vulnerabilities of Passwords

Exploiting multiple password use

- Attacks can also become much more effective or damaging if different network devices share the same or a similar password for a given user.
- Countermeasures: policy that forbids the same or similar password on particular network devices.

Electronic monitoring

- Passwords communicated across a network to log on to a remote system is vulnerable to eavesdropping.
- Simple encryption will not fix this problem, because the encrypted password is, in effect, the password and can be observed and reused by an adversary

Multi-factor Authentication

Using A combination of more than one method, such as token and password (or personal identification number [PIN] or token and biometric device)

Two-factor authentication is a security process in which the user provides **two** means of identification, one of which is typically a physical token, such as a card, and the other of which is typically something memorized, such as a security code

Single Sign-On (SSO)

SSO can generally be defined as the process for consolidating all organization platform-based administration, authentication and authorization functions into a single centralized administrative function. This function would provide the appropriate interfaces to the organization's information resources, which may include:

- Client-server and distributed systems
- Mainframe systems
- Network security including remote access mechanisms

SSO Advantages

- Multiple passwords are no longer required; therefore, a user may be more inclined and motivated to select a stronger password.
- It improves an administrator's ability to manage users' accounts and authorizations to all associated systems.
- It reduces administrative overhead in resetting forgotten passwords over multiple platforms and applications.
- It reduces the time taken by users to log into multiple applications and platforms.

SSO Disadvantages

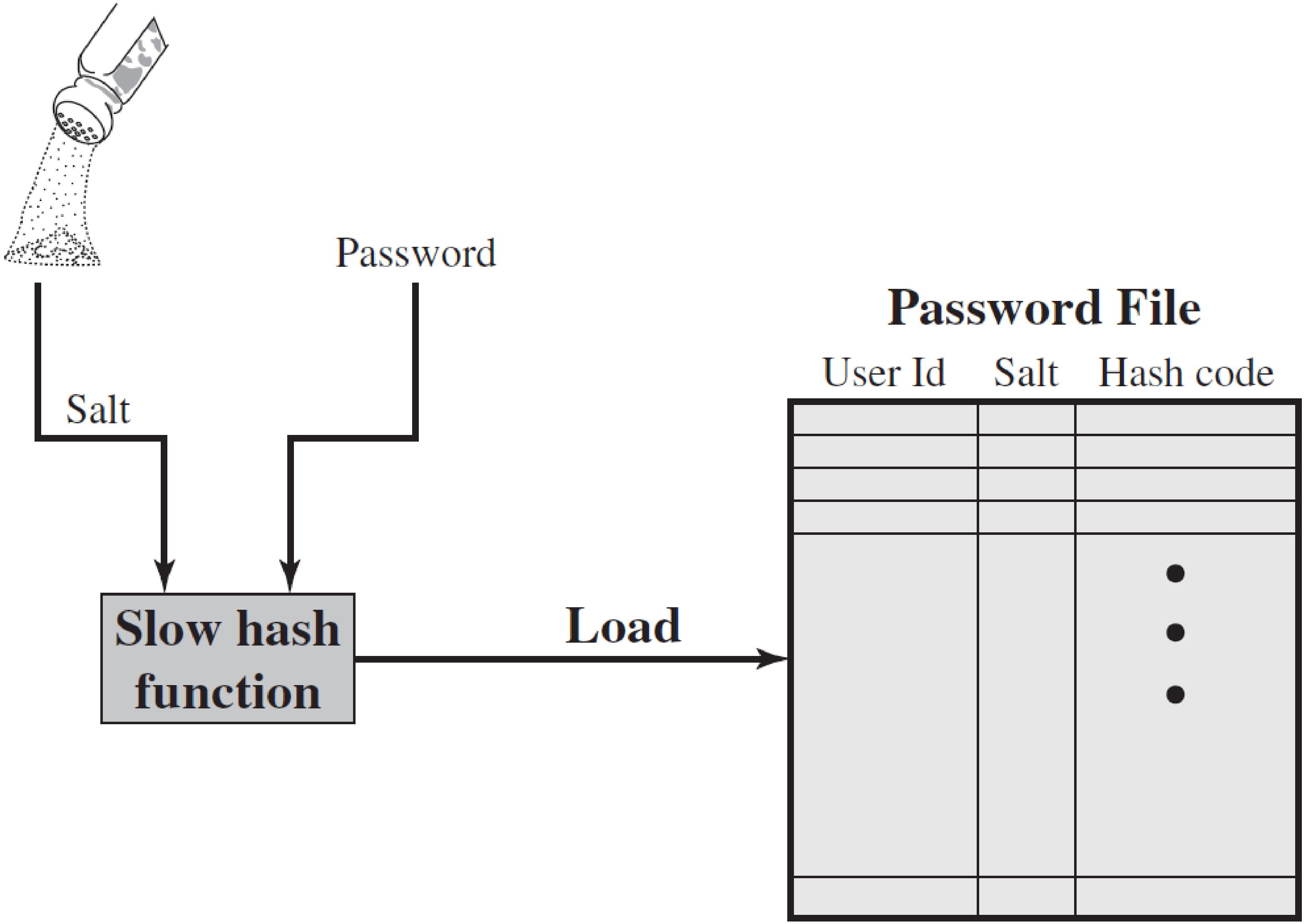
- Support for all major operating system environments is difficult. SSO implementations will often require a number of solutions integrated into a total solution for an enterprise's IT architecture.
- The costs associated with SSO development can be significant when considering the nature and extent of interface development and maintenance that may be necessary.
- The centralized nature of SSO presents the possibility of a single point of failure and total compromise of an organization's information assets. For this reason, "strong authentication“ in the form of complex password requirements and the use of biometrics is frequently implemented.

Hashed passwords with salt

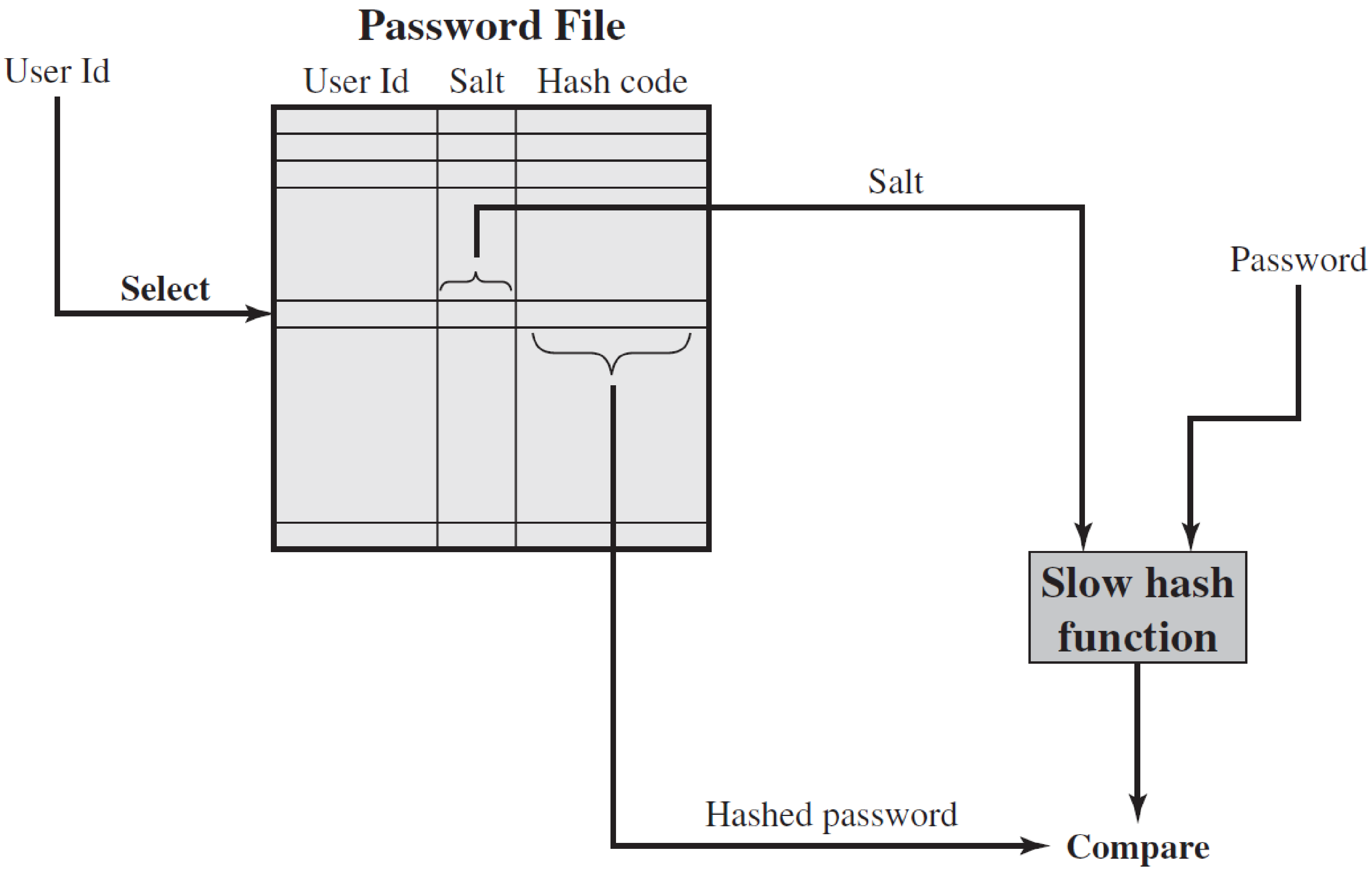
- Most systems, e.g., Linux, store hashed passwords and a salt value for better security
- Steps to store a password:
  - Given a password (selected by user or assigned by system), the system generates a fixed length pseudorandom/random number, called salt
  - Older system uses time when the password is created to generate the salt
  - Use hash function to generate a fixed length hashed code of the password and its salt
  - Store the hashed code and a plaintext copy of the salt in the password file

Hashed passwords with salt

- Steps to verify a password:
  - Given a user ID and a password, the system uses the ID to retrieve the plaintext salt and the encrypted password
  - Use the salt and the supplied password as input to the encryption function
  - If the result matches the stored encrypted value, the password is accepted



(a) Loading a new password



(b) Verifying a password

Purposes of Using Salt

- **To prevents duplicate passwords in the password file**

  Each password is assigned a different salt value. Thus even if two users use the same password, the stored hashed passwords would be different

- **To significantly increases the difficulty of offline dictionary attacks**

  A b bit salt will increase the number of possible passwords by a factor of 2b, and thus guessing password would be harder

- **To makes almost impossible to find out if a person use the same password on two or more systems**

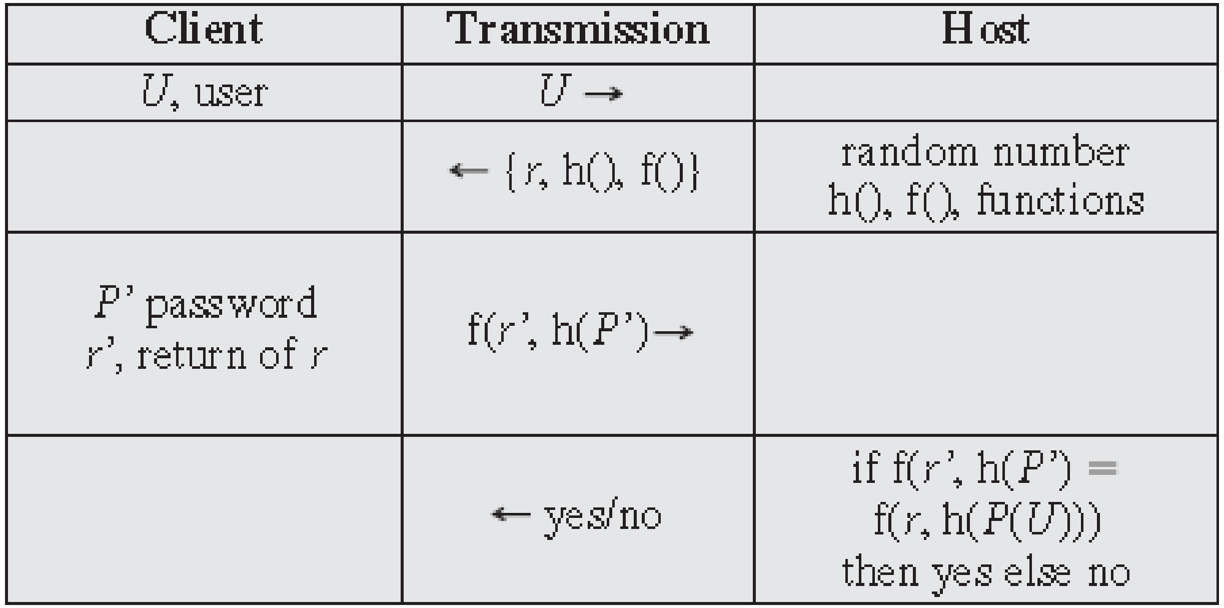
Remote user authentication

- Remote user authentication raises additional security threats such as eavesdropping and replay attack
  - The counter measure generally relies on challenge-response protocol, such as Kerberos

Challenge-response protocol

- Steps of a simple challenge-response protocol
- User transmits his/her ID to the remote host
- The host generates a random number r, called a nonce, and returns it to the user. The host also specifies two functions, a hash function h() and f() to be used for the user’s response
  - The host keeps function h() for the password of each of its users U ⍰ h(P(U))
  - This is the challenge
- The user must send a correct response f(r’, h(P’)) the host
  - r’=r and P’ is the user’s password
- The host calculates f(r, h(P(U)) and compares it with the received f(r’, h(P’))

Challenge-Response Protocol



Client	Transmission	Host
U, user	$U \rightarrow$	
	$\leftarrow \{r, \mathrm{h}(), \mathrm{f}()\}$	random number h(), f(), functions
P' password r', return of r	$\mathrm{f}(r', \mathrm{h}(P') \rightarrow$	
	$\leftarrow$ yes/no	if $\mathrm{f}(r', \mathrm{h}(P') =$ $\mathrm{f}(r, \mathrm{h}(P(U)))$ then yes else no

(a) Protocol for a password

- From W. Stallings and L. Brown, “Computer Security, Principles and Practice, 2nd edition

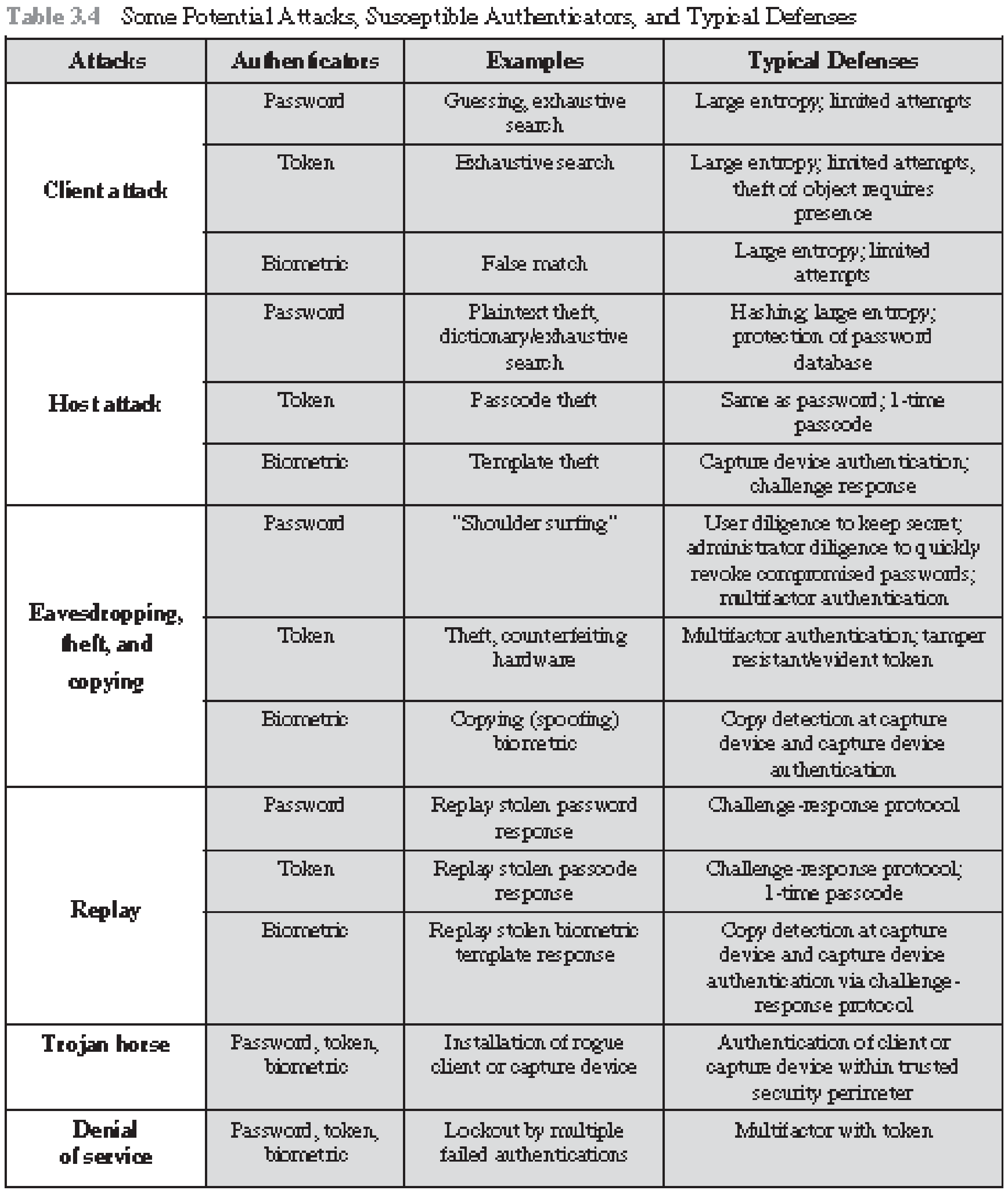


Table 3.4 Some Potential Attacks, Susceptible Authenticators, and Typical Defenses

Attacks	Authenticators	Examples	Typical Defenses
Client attack	Password	Guessing, exhaustive search	Large entropy; limited attempts
	Token	Exhaustive search	Large entropy; limited attempts, theft of object requires presence
	Biometric	False match	Large entropy; limited attempts
Host attack	Password	Plaintext theft, dictionary/exhaustive search	Hashing; large entropy; protection of password database
	Token	Passcode theft	Same as password; 1-time passcode
	Biometric	Template theft	Capture device authentication; challenge response
Eavesdropping, theft, and copying	Password	"Shoulder surfing"	User diligence to keep secret; administrator diligence to quickly revoke compromised passwords; multifactor authentication
	Token	Theft, counterfeiting hardware	Multifactor authentication; tamper resistant/evident token
	Biometric	Copying (spoofing) biometric	Copy detection at capture device and capture device authentication
Replay	Password	Replay stolen password response	Challenge-response protocol
	Token	Replay stolen passcode response	Challenge-response protocol; 1-time passcode
	Biometric	Replay stolen biometric template response	Copy detection at capture device and capture device authentication via challenge-response protocol
Trojan horse	Password, token, biometric	Installation of rogue client or capture device	Authentication of client or capture device within trusted security perimeter
Denial of service	Password, token, biometric	Lockout by multiple failed authentications	Multifactor with token

- From W. Stallings and L. Brown, "Computer Security, Principles and Practice, 2nd edition

Password Selection Strategies

- User education
- Computer-generated passwords
- Reactive password checking
- Proactive password checking

Q & A?

End of Lecture 09

SLIIT
FACULTY OF COMPUTING