3.1 User Authentication

RFC 4949 defines user authentication as: the process of verifying an identity claimed by or for a system entity

Authentication Process

Authentication process

- Fundamental building block and primary line of defense
- Basis for access control and user accountability

Two main steps:

- Identification step: presenting an identifier to the security system
- Verification step: presenting or generating authentication info

Means of User Authentication

Four means of authenticating user's identity, based on

- Something you have - USB
- Something you know - password
- Something you are - fingerprint
- Somebody you know - social network of user

All of these have their own issues

Should use Multi-Factor Authentication (MFA)

Text Password Security Problems

Reality:

- People create weak passwords
- People share passwords (between multiple accounts)
- Insecurely store passwords
- Fall victim to phishing attacks

Vulnerabilities:

- **Offline** dictionary attack
- **Specific** account attack
- **Popular** password attack
- Workstation **hijacking**
- Exploiting user **mistakes**
- Exploiting password **reuse**
- Insecure **transmission**

Countermeasures

- Stop unauthorized access to **password file**
- **Intrusion detection** measures
- Account **lockout** mechanisms
- Training & enforcement of **policies**
- Automatic workstation **logout**
- **Encrypted** network links

Use of Hashed Passwords at the Server-Side

Loading a new password

> Salt + Password -> (HASH FUNCTION) -> Load hash code to database

Verifying password

> Salt + Password -> (HASH FUNCTION) -> Compare genereated hash code with saved hash code in database

Password Cracking Approaches

Dictionary attacks: try each possible password then obvious variants in large dictionary against hash in password file

Rainbow table attacks:

- Precompute tables of hash values of all possible passwords
- A mammoth table of hash values
- 1.4GB table cracks 99.9% of alphanumeric Windows passwords in 13.8 secs
- Not feasible if larger salt values used

Password File Access Control

Can block offline guessing attacks by denying access to hashed passwords

- Make available only to privileged users
- Often using a separate (from users IDs) shadow password

Still have vulnerabilities

- Exploit OS bugs
- Error on permissions making it readable
- Users reuse same password on other systems
- Access from unprotected backup media
- Sniff passwords in unprotected network traffic

Using Better Passwords

Techniques:

- User education
- Reactive password checking (having a password and noticing it could be stronger)
- Proactive password checking (When thinking of a password make sure its strong)

Reactive password:

- System periodically runs its own password cracked to find guessable passwords
- Websites that tell you if a password is strong or not

Proactive password checking:

- Rule enforcement: special characters, at least 7 characters, etc...
- Password cracker: reject weak passwords
- Markov model: generate guessable passwords, reject the generated ones
- Bloom filter: Build tables based on dictionary using hashes, check desired passwords against the table
  - It uses hashes of existing passwords, and compare with your password. This gives us a probability distribution to show if there is a really common password being used.

Other solutions:

- Graphical passwords: security and usability concerns

- Password hashing systems: security and usability concerns
- Single sign-on systems: business model limitations, security concern
- Browser-based password managers: Easy to use, potentially protect against phishing

Biometric Authentication

Biometric authentication: authenticating users based on one of their physical characteristics

- Hand, signature, face, voice, retina, finger, iris

Operation of biometric systems:

- Enrollment: sensor -> feature extractor -> store in database
- Verification: sensor -> feature extractor -> feature matcher with database
- Identification: see if feature extractor shows up in database

Biometric accuracy:

- Never gets identical templates
- Problems with false match / false non-match

Kerobos

Authentication service that provides a centralized mutual authentication in a distributed network.

- Allow users access to distributed service in the network
- A workstation cannot be trusted to identify its user
- Rather all trust a center authentication server
- Relies exclusively on symmetric encryption
- Requires a user to prove his or her identity for each service invoked, also requires servers to prove their identity its user

Web Single Sign-On (SSO) Phishing

A website that asks for you to sign on with your email, however it will just hack your email