Computer and Network Security: Authentication Protocols: Human Authentication

Kameswari Chebrolu

Recap: Basic Idea

- Long term key:
  - Could be shared/secret key or public key
  - Exists before authentication protocol begins
  - Does not change from session to session
- Short term / Session key:
  - Often shared key because public key operations are expensive
  - **Authentication protocols** help agree on this key
  - Valid only for that session of the protocol

Background

- Authentication started out as Human–computer interaction
  - Password based logins
  - Many drawbacks (will see soon)
- Overcome via cryptographic protocols
  - Prove identity by performing a cryptographic operation (hash, encryption etc)
  - Supports both human-computer and computer-computer interaction

Outline

- **Human Authentication**
  - Focus: Password based systems
- Cryptographic Authentication (Human as well as computer)
  - One way authentication (shared and public key)
  - Mutual authentication (shared and public key)
  - Mediated authentication (shared key)
  - How to incorporate session key exchange?
  - How to follow it up to provide privacy and Integrity?

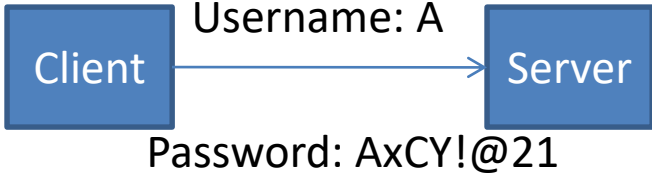
Human Authentication

- Use cases: Web Logins, high-security areas etc
- Mechanisms:
  - **What user knows (password, pin)**
  - What user has (physical key, card)
  - What user is (fingerprint, retina)

Password based Authentication

Security Risks

- Eavesdropping
  - While typing, clear text transmission
- Client-side malware
  - Keystroke capture
- Phishing (fake website)
- Guessing (easy passwords)
- Server side break-in/offline cracking
  - Get list of stored passwords



Risk Mitigation

- Eavesdropping
  - Display **** while typing; Send encrypted password
- Client-side malware
  - Very difficult to detect
  - Two-factor authentication helps (password + pin got via SMS)
- Phishing: To be covered later in web security

- Guessing
  - Lock account after X failed guesses (opens up DOS attack)
  - Captcha to discourage automated guessing
  - Encourage use of strong passwords
    - Atleast 8 characters, one numeric, one special symbol
    - Change password every month; password cannot be same as last N passwords
    - Password meter (strength)
- How long should the password be? 64 bit of randomness → 11 characters

Server side Break-in

- How should the website store the passwords?
  - Cleartext no good if website hacked
- Store hash of each password: H(password)
  - Subject to offline password guessing
    - E.g. SHA256 is used (1 billion guesses per sec), cracking half of 100 million accounts can take minutes to hours
  - Use salt i.e. for a user store “s, H(password|s)”
    - Increases workload (maybe to days)
  - Slow the hash i.e. for a user store “s, H(H(….H(password|s)….))”

Conclusions

- Human password based authentication has many drawbacks
- Looked at commonly used mitigation techniques
- Best practice: Two factor authentication and Cryptographic protocols