CS 155 May 4, 2004

User Authentication

John Mitchell

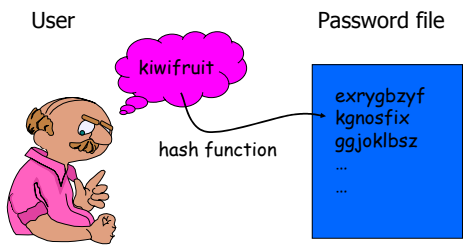
Outline

- User authentication
  - Password authentication, salt
  - Challenge-Response
  - Biometrics
  - Token-based authentication
- Authentication in distributed systems
  - Single sign-on, Microsoft Passport
  - File sharing examples
    - NFS, SMB, LanMan, NTLM
  - Kerberos

Password authentication

- Basic idea
  - User has a secret password
  - System checks password to authenticate user
- Issues
  - How is password stored?
  - How does system check password?
  - How easy is it to guess a password?
    - Difficult to keep password file secret, so best if it is hard to guess password even if you have the password file

Basic password scheme



Basic password scheme

- Hash function h : strings → strings
  - Given h(password), hard to find password
  - No known algorithm better than trial and error
- User password stored as h(password)
- When user enters password
  - System computes h(password)
  - Compares with entry in password file
- No passwords stored on disk

Unix password system

- Hash function is 25xDES
  - Number 25 was meant to make search slow
- Password file is publicly readable
  - Other information in password file …
- Any user can try "dictionary attack"
  - User looks at password file
  - Computes hash(word) for every word in dictionary
- "Salt" makes dictionary attack harder

R.H. Morris and K. Thompson, Password security: a case history, Communications of the ACM, November 1979

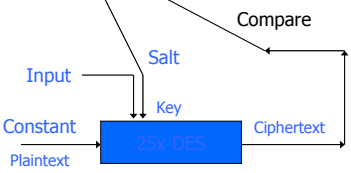
Dictionary Attack – some numbers

- Typical password dictionary
  - 1,000,000 entries of common passwords
    - people's names, common pet names, and ordinary words.
  - Suppose you generate and analyze 10 guesses per second
    - This may be reasonable for a web site; offline is *much* faster
  - Dictionary attack in at most 100,000 seconds = 28 hours, or 14 hours on average
- If passwords were random
  - Assume six-character password
    - Upper- and lowercase letters, digits, 32 punctuation characters
    - 689,869,781,056 password combinations.
    - Exhaustive search requires 1,093 years on average

Salt

- Password line

walt:fURfuu4.4hY0U:129:129:Belgers:/home/walt:/bin/csh

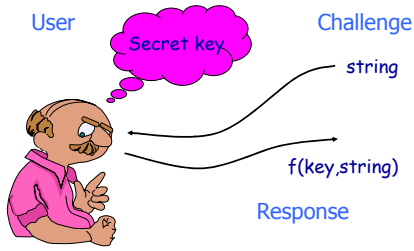


When password is set, salt is chosen randomly
12-bit salt slows dictionary attack by factor of 2^{12}

Advantages of salt

- Without salt
  - Same hash functions on all machines
    - Compute hash of all common strings once
    - Compare hash file with all known password files
- With salt
  - One password hashed 2^{12} different ways
    - Precompute hash file?
      - Need much larger file to cover all common strings
    - Dictionary attack on known password file
      - For each salt found in file, try all common strings

Challenge-response



Challenge-response authentication

- Challenge
  - System presents user with some string
- Response
  - User computes f(key,string)
- Authentication
  - Check property of f(key,string)
  - Secret data can stay secret: no password is sent

Why is this *much* better over a network?

Biometrics

- Use a person's physical characteristics
  - fingerprint, voice, face, keyboard timing, …
- Advantages
  - Cannot be disclosed, lost, forgotten
- Disadvantages
  - Cost, installation, maintenance
  - Reliability of comparison algorithms
    - False positive: Allow access to unauthorized person
    - False negative: Disallow access to authorized person
  - Privacy?
  - If forged, how do you revoke?

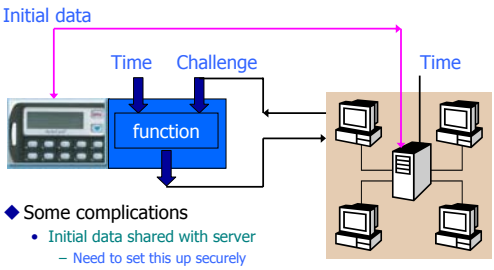
Biometrics

- Common uses
  - Specialized situations, physical security
  - Combine
    - Multiple biometrics
    - Biometric and PIN
    - Biometric and token

Token-based authentication

- Several configurations and modes of use
  - Device produces password, user types into system
  - User unlocks device using PIN
  - User unlocks device, enters challenge
- Example: S/Key
  - User enters string, devices computes sequence
    - p_0 = hash(string|rand); p_{i+1} = hash(p_i)
    - p_n placed on server; set counter k = n
  - Device can be used n times before reinitializing
    - Send p_{k-1} = to server, set k = k-1
    - Sever checks hash(p_{k-1}) = p_k , stores p_{k-1}

Other methods (several vendors)



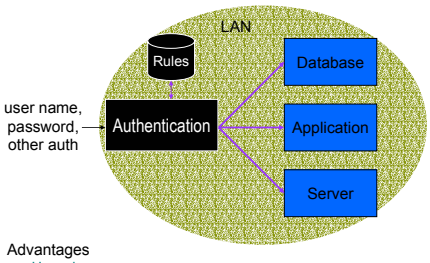
- Some complications
  - Initial data shared with server
    - Need to set this up securely
    - Shared database for many sites
  - Clock skew

Authentication in Distributed Systems

- Single sign-on
- Web identity
  - Passport
  - Liberty Alliance
- Remote file systems
- Kerberos

Single sign-on systems

e.g. Securant, Netegrity, Oblix



- Advantages
  - User signs on once
  - No need for authentication at multiple sites, applications
  - Can set central authorization policy for the enterprise

Microsoft Passport

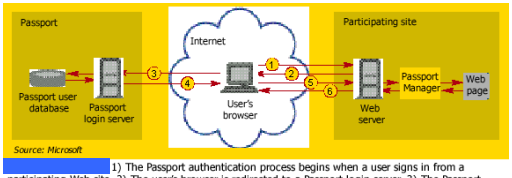
- Launched 1999
  - Claim > 200 million accounts in 2002
  - Over 3.5 billion authentications each month
- Log in to many websites using one account
  - Used by MS services Hotmail, MSN Messenger or MSN subscriptions; also Radio Shack, etc.
  - Hotmail or MSN users automatically have Microsoft Passport accounts set up
- Passport may continue to evolve; bugs have been uncovered

www.networkmagazine.com/article/NMG20020304S0003

Four parts of Passport account

- Passport Unique Identifier (PUID)
  - Assigned to the user when he or she sets up the account
- User profile, required to set up account
  - Phone number or Hotmail or MSN.com e-mail address
  - Also name, ZIP code, state, or country, …
- Credential information
  - E-mail address or phone number
  - Minimum six-character password or PIN
  - Four-digit security key, used for a second level of authentication on sites requiring stronger sign-in credentials
- Wallet
  - Passport-based application at passport.com domain
  - E-commerce sites with Express Purchase function use wallet information rather than prompt the user to type in data

Passport log-in



1) The Passport authentication process begins when a user signs in from a participating Web site. 2) The user's browser is redirected to a Passport login server. 3) The Passport server prompts the user for an e-mail address and password and checks it against its database. 4) The Passport server places three encrypted cookies to the user's browser, which is redirected to the participating Web site. 5) The participating site decrypts the authentication cookie via the Passport Manager object. 6) The site serves up the requested Web page.

Some Problems (now fixed)

- Request password reset May 8, 2003
  - A remote user can use the .NET Passport password reset form to request that an arbitrary user's password be changed
  - Example https://register.passport.net/emailpwdreset.srf?lc=1033&em=victim@hotmail.com&id=&cb=&prefem=attacker@attacker.com&rst
- Complete via email with link
  - The remote user (attacker@attacker.com) receives e-mail from Passport providing a URL to change the password
  - The form does not require the previous password
  - See: http://securitytracker.com/alerts/2003/May/1006728.html , http://securityfocus.com/archive/75/320768/2003-05-05/2003-05-11/0

Another problem

Marc Slemko's Passport exploit, since fixed by Microsoft

- Description
  - Hotmail user stores credit card info in Wallet
  - User logs into Hotmail
  - Within 15 min of logging in, user reads an email
  - Email author steals all the info in Wallet
  - User has no indication that attack occurred
- How it works
  - Exploits Passport cookie-based authentication
  - Hotmail has web interface, processes html
    - Can embed html that contacts another domain
  - See http://alive.znep.com/~marcs/passport/

Liberty Alliance

- Open-standard alternative to Passport
  - Sun Microsystems, AOL, MasterCard, Bank of America, Sony, General Motors, …
  - P3P privacy promises
- Goals
  - Single sign-on using a federated network identity architecture
  - Give users choice and control over their personal information
  - Open industry standards
- Demo March 2004
  - See http://www.projectliberty.org/ for more info