Security

Generating Random Passwords

- Random (pseudorandom) number generator period critical!
- Example: PDP-11 randomly generated passwords of length 8, and composed of capital letters and digits
  - Number of possible passwords = $(26 + 10)^8 = 36^8 = 2.8 \times 10^{12}$
  - Took 0.00156 to test a password, so would take about 140 years to try all
- Attacker noticed the pseudorandom number generator on PDP-11, with word size of 16 bits, had period of $2^{16} – 1$
  - Number of possible passwords = $2^{16} – 1 = 65{,}535 = 6.5 \times 10^4$
  - Took 0.00156 to test a password, so would take about 102 seconds to try all
- When launched, found all passwords in under 41 seconds

Remembering Random Passwords

- Humans can repeat with perfect accuracy 8 meaningful items
  - Like digits, letters, words
- Write them down
  - Put them in a place where others are unlikely to get to them
  - Purse or wallet is good; keyboard or monitor is not
- Write obscured versions of passwords
  - Let $p \in P$ be password; choose invertible transformation algorithm t: $P \rightarrow A$
  - Write down $t^{-1}(p)$ but not t
  - Now user must memorize t, not each individual password
- Use a password manager (password wallet)
  - Now must remember password to unlock the other passwords

Pronounceable Passwords

- Generate phonemes randomly
  - Phoneme is unit of sound, eg. *cv*, *vc*, *cvc*, *vcv*
  - Examples: helgoret, juttelon are; przbqxdfl, zxrptglfn are not
- Problem: too few
- Solution: key crunching
  - Run long key through hash function and convert to printable sequence
  - Use this sequence as password
- Bigger problem: distribution of passwords
  - Probabilities of selection of particular phonemes, hence passwords, not equiprobable
  - Generated passwords tend to cluster; if an attacker finds a cluster with passwords user is likely to select, this reduces search space greatly

User Selection

- Problem: people pick easy to guess passwords
  - Based on account names, user names, computer names, place names
  - Dictionary words (also reversed, odd capitalizations, control characters, “elite-speak”, conjugations or declensions, swear words, Torah/Bible/Koran/… words)
  - Too short, digits only, letters only
  - License plates, acronyms, social security numbers
  - Personal characteristics or foibles (pet names, nicknames, job characteristics, *etc*.

Picking Good Passwords

- “WtBvStHbChCsLm?TbWtF.+FSK”
  - Intermingling of letters from Star Spangled Banner , some punctuation, and author’s initials
- What’s good somewhere may be bad somewhere else
  - “DCHNH,DMC/MHmh” bad at Dartmouth (“Dartmouth College Hanover NH, Dartmouth Medical Center/Mary Hitchcock memorial hospital”), ok elsewhere (probably)
- Why are these now bad passwords? ☹

Passphrases

- A password composed of multiple words and, possibly, other characters
- Examples:
  - “home country terror flight gloom grave”
    - From Star Spangled Banner, third verse, third and sixth line
  - “correct horse battery staple”
    - From *xkcd* (https://xkcd.com/936)
- Caution: the above are no longer good passphrases

Remembering Passphrases

- Memorability is good example of how environment affects security
  - Study of web browsing shows average user has 6-7 passwords, sharing each among about 4 sites (from people who opted into a study of web passwords)
    - Researchers used an add-on to a browser that recorded information about the web passwords but *not* the password itself
- Users tend not to change password until they know it has been compromised
  - And when they do, the new passwords tend to be as short as allowed
- Passphrases seem as easy to remember as passwords
  - More susceptible to typographical errors
  - If passphrases are text as found in normal documents, error rate drops

Password Manager (Wallet)

- A mechanism that encrypts a set of user's passwords
- User need only remember the encryption key
  - Sometimes called "master password"
  - Enter it, and then you can access all other passwords
- Many password managers integrated with browsers, cell phone apps
  - So you enter the master password, and password manager displays the appropriate password entry
  - When it does so, it shows what the password logs you into, such as the institution with the server, and hides the password; you can then have it enter the password for you

Salting

- Goal: slow dictionary attacks
- Method: perturb hash function so that:
  - Parameter controls *which* hash function is used
  - Parameter differs for each password
  - So given *n* password hashes, and therefore *n* salts, need to hash guesses *n* times

Examples

- Vanilla UNIX method
  - Perturb a table in the cipher in one of 4096 ways
  - Use modified cipher to encipher 0 message with password as key; iterate 25 times
- Linux method
  - Generate salt randomly
  - Hash password (usually with SHA-256)
  - Prepend salt to get string to be stored

Password Aging

- Force users to change passwords after some time has expired
  - How do you force users not to re-use passwords?
    - Record previous passwords
    - Block changes for a period of time
  - Give users time to think of good passwords
    - Don’t force them to change before they can log in
    - Warn them of expiration days in advance

Pass Algorithms

- Challenge-response with the function *f* itself a secret
  - Example:
    - Challenge is a random string of characters such as “abcdefg”, “ageksido”
    - Response is some function of that string such as “bdf”, “gkip”
  - Can alter algorithm based on ancillary information
    - Network connection is as above, dial-up might require “aceg”, “aesd”
  - Usually used in conjunction with fixed, reusable password

One-Time Passwords

- Password that can be used exactly *once*
  - After use, it is immediately invalidated
- Challenge-response mechanism
  - Challenge is number of authentications; response is password for that particular number
- Problems
  - Synchronization of user, system
  - Generation of good random passwords
  - Password distribution problem

Hardware Support

- Token-based
  - Used to compute response to challenge
    - May encipher or hash challenge
    - May require PIN from user
- Temporally-based
  - Every minute (or so) different number shown
    - Computer knows what number to expect when
  - User enters number and fixed password

Biometrics

- Automated measurement of biological, behavioral features that identify a person
  - Fingerprints: optical or electrical techniques
    - Maps fingerprint into a graph, then compares with database
    - Measurements imprecise, so approximate matching algorithms used
  - Voices: speaker verification or recognition
    - Verification: uses statistical techniques to test hypothesis that speaker is who is claimed (speaker dependent)
    - Recognition: checks content of answers (speaker independent)

Other Characteristics

- Can use several other characteristics
  - Eyes: patterns in irises unique
    - Measure patterns, determine if differences are random; or correlate images using statistical tests
  - Faces: image, or specific characteristics like distance from nose to chin
    - Lighting, view of face, other noise can hinder this
  - Keystroke dynamics: believed to be unique
    - Keystroke intervals, pressure, duration of stroke, where key is struck
    - Statistical tests used

Cautions

- These can be fooled!
  - Assumes biometric device accurate *in the environment it is being used in!*
  - Transmission of data to validator is tamperproof, correct

Location

- If you know where user is, validate identity by seeing if person is where the user is
  - Requires special-purpose hardware to locate user
    - GPS (global positioning system) device gives location signature of entity
    - Host uses LSS (location signature sensor) to get signature for entity

Multiple Methods

- Example: “where you are” also requires entity to have LSS and GPS, so also “what you have”
- Can assign different methods to different tasks
  - As users perform more and more sensitive tasks, must authenticate in more and more ways (presumably, more stringently) File describes authentication required
    - Also includes controls on access (time of day, *etc*.), resources, and requests to change passwords
  - Pluggable Authentication Modules

PAM

- Idea: when program needs to authenticate, it checks central repository for methods to use
- Library call: *pam_authenticate*
  - Accesses file with name of program in */etc/pam_d*
- Modules do authentication checking
  - *sufficient*: succeed if module succeeds
  - *required*: fail if module fails, but all required modules executed before reporting failure
  - *requisite*: like *required*, but don't check all modules
  - *optional*: invoke only if all previous modules fail

Example PAM File

```
auth sufficient /usr/lib/pam_ftp.so

auth required   /usr/lib/pam_unix_auth.so use_first_pass

auth required   /usr/lib/pam_listfile.so onerr=succeed \
     item=user sense=deny file=/etc/ftpusers
```

For ftp:

1. If user “anonymous”, return okay; if not, set PAM_AUTHTOK to password, PAM_RUSER to name, and fail
2. Now check that password in PAM_AUTHTOK belongs to that of user in PAM_RUSER; if not, fail
3. Now see if user in PAM_RUSER named in /etc/ftpusers; if so, fail; if error or not found, succeed

Users

- Exact representation tied to system
- Example: Linux/UNIX systems
  - Login name: used to log in to system
    - Logging usually uses this name
  - User identification number (UID): unique integer assigned to user
    - Kernel uses UID to identify users
    - One UID per login name, but multiple login names may have a common UID

Multiple Identities

- Linux/UNIX systems again
  - Real UID: user identity at login, but changeable
  - Effective UID: user identity used for access control
    - Setuid changes effective UID
  - Saved UID: UID before last change of UID
    - Used to implement least privilege
    - Work with privileges, drop them, reclaim them later
  - Audit/Login UID: user identity used to track original UID
    - Cannot be altered; used to tie actions to login identity

Groups

- Used to share access privileges
- First model: alias for set of principals
  - Processes assigned to groups
  - Processes stay in those groups for their lifetime
- Second model: principals can change groups
  - Rights due to old group discarded; rights due to new group added

Roles

- Group with membership tied to function
  - Rights given are consistent with rights needed to perform function
- Uses second model of groups
- Example: DG/UX
  - User *root* does not have administration functionality
  - System administrator privileges are in *sysadmin* role
  - Network administration privileges are in *netadmin* role
  - Users can assume either role as needed

Network Security

- Messages in transit: use encryption
- Link encryption
  - Each host enciphers message so host at “next hop” can read it
  - Message can be read at intermediate hosts
- End-to-end encryption
  - Host enciphers message so host at other end of communication can read it
  - Message cannot be read at intermediate hosts

Examples

- SSH protocol
  - Messages between client, server are enciphered, and encipherment, decipherment occur only at these hosts
  - End-to-end protocol
- PPP Encryption Control Protocol
  - Host gets message, deciphers it
    - Figures out where to forward it
    - Enciphers it in appropriate key and forwards it
  - Link protocol

Cryptographic Considerations

- Link encryption
  - Each host shares key with neighbor
  - Can be set on per-host or per-host-pair basis
    - Windsor, stripe, seaview each have own keys
    - One key for (windsor, stripe); one for (stripe, seaview); one for (windsor, seaview)
- End-to-end
  - Each host shares key with destination
  - Can be set on per-host or per-host-pair basis
  - Message cannot be read at intermediate nodes

Firewalls

- Host that mediates access to a network
  - Allows, disallows accesses based on configuration and type of access
- Example: block Conficker worm
  - Conficker connects to botnet, which can use system for many purposes
    - Spreads through a vulnerability in a particular network service
  - Firewall analyze packets using that service remotely, and look for Conficker and its variants
    - If found, packets discarded, and other actions may be taken
  - Conficker also generates list of host names, tried to contact botnets at those hosts
    - As set of domains known, firewall can also block outbound traffic to those hosts

Filtering Firewalls

- *A*ccess control based on attributes of packets and packet headers
  - Such as destination address, port numbers, options, etc.
  - Also called a *packet filtering firewall*
  - Does not control access based on content
  - Examples: routers, other infrastructure systems

Proxy

- Intermediate agent or server acting on behalf of endpoint without allowing a direct connection between the two endpoints
  - So each endpoint talks to proxy, thinking it is talking to other endpoint
  - Proxy decides whether to forward messages, and whether to alter them

Proxy Firewall

- Access control done with proxies
  - Usually bases access control on content as well as source, destination addresses, etc.
  - Also called an *applications level* or *application level firewall*
  - Example: virus checking in electronic mail
    - Incoming mail goes to proxy firewall
    - Proxy firewall receives mail, scans it
    - If no virus, mail forwarded to destination
    - If virus, mail rejected or disinfected before forwarding

Example

- Want to scan incoming email for malware
- Firewall acts as recipient, gets packets making up message and reassembles the message
  - It then scans the message for malware
  - If none, message forwarded
  - If some found, mail is discarded (or some other appropriate action)
- As email reassembled at firewall by a mail agent acting on behalf of mail agent at destination, it's a proxy firewall (application layer firewall)

Stateful Firewall

- Keeps track of the state of each connection
- Similar to a proxy firewall
  - No proxies involved, but this can examine contents of connections
  - Analyzes each packet, keeps track of state
  - When state indicates an attack, connection blocked or some other appropriate action taken