

Password Authentication & Protection



Passwords

Why Are They Important?

- Passwords are cheap to deploy, but also act as the first line of defense in a security arsenal.
 - They are also often the weakest link.
- Examples of what they protect:
 - ATMs and bank accounts
 - Nuclear power and other critical infrastructure systems
 - Company proprietary information and systems
 - Email accounts (Gmail, Hotmail, Yahoo, AOL, etc.)
 - Student information (e.g. MyUalbany & WebCT)

Passwords

Authentication

- Passwords have been used for centuries, e.g. guards and sentries
- Passwords = secret authentication code used for access.
- Answers the question: How do you prove to someone that you are who you claim to be?
- Authentication methods:
 - What you know (Passwords, Secret keys)
 - Where you are (IP Addresses)
 - What you are (Biometrics)
 - What you have (Secure tokens)



Passwords

AAA of Password Security

- Authentication (& Identification)
 - Establishes that the user is who they say they are (credentials).
- Authorization
 - The process used to decide if the authenticated person is allowed to access specific information or functions.
- Access Control
 - Restriction of access (includes authentication & authorization)

Passwords

How Can Passwords Be Stored?



Filing System
Clear text



Dedicated Authentication Server
Clear text



Encrypted
Password + Encryption = bf4ee8HjaQkbw



Hashed
Password + Hash function =
aad3b435b51404eeaad3b435b51404ee

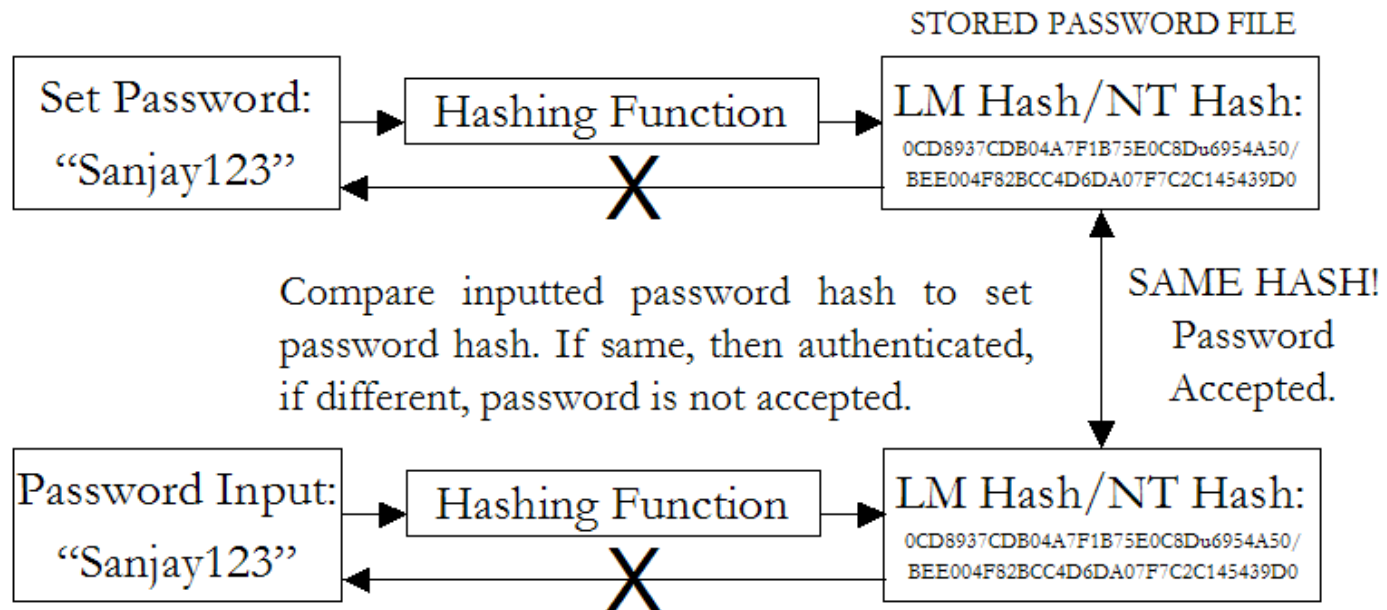


Salted Hash
(Username + Salt + Password) + Hash function =
e3ed2cb1f5e0162199be16b12419c012

Passwords

How Are Passwords Stored? - Hashing

- Usually stored as hashes (not plain text)
 - Plain-text is converted into a message digest through use of a hashing algorithm (i.e. MD5, SHA)



Passwords

How Are Passwords Stored? - Hashing

- Hash function H must have some properties:
 - **One-way:** given $H(\text{password})$, hard to find password
 - No known algorithm better than trial and error
 - **Collision-resistant:** given $H(\text{password1})$, hard to find password2 such that: $H(\text{password1}) = H(\text{password2})$
 - It should even be hard to find any pair $p1, p2$ s.t.
 $H(p1) = H(p2)$

Passwords

How Are Passwords Stored? – Early UNIX Systems

- In past UNIX systems, password used modified DES (encryption algorithm) as if it were a hash function
 - Encrypts NULL string using password as the key (truncates passwords to 8 characters!)
 - Caused artificial slowdown: ran DES 25 times
- Also stored password file in directory: `/etc/passwd/`
 - World-readable (anyone who accessed the machine would be able to copy the password file to crack at their leisure)
 - Contained userIDs/groupIDs used by many system programs
 - Can instruct modern UNIXes to use MD5 hash function

Passwords

Plain Text Security Issues

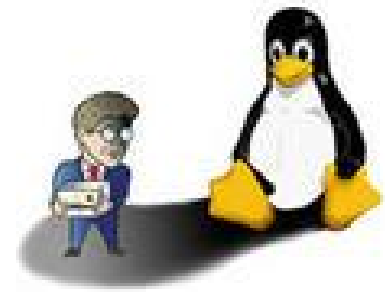
System administrator at MIT was editing the password file and another was editing the daily message (appeared on everyone's login terminal). Due to a software error, the editor files were switched and the password file was printed every time someone logged in.

- Robert Morris & Ken Thompson (April 3, 1978)

Passwords

How Are Passwords Stored? - Newer UNIX Systems

- Password hashes stored in /etc/shadow directory (or similar)
 - only readable by system administrator (root)
- Less sensitive information still in /etc/passwd
- Added expiration dates for passwords
- Early “shadow” implementations on Linux called the login program which had a buffer overflow!



Passwords

How Are Passwords Stored? – Windows NT/2k/XP/Vista

- Uses 2 functions for “hashing” passwords:
 1. LAN Manager hash (LM hash)
 - Password is padded with zeros until there are 14 characters.
 - It is then converted to uppercase and split into two 7-character pieces
 - Each half is encrypted using an 8-byte DES (data encryption standard) key
 - Result is combined into a 16-byte, one way hash value
 2. NT hash (NT hash)
 - Converts password to Unicode and uses MD4 hash algorithm to obtain a 16-byte value
- Hashes stored in Security Accounts Manager (SAM)
 - Locked within system kernel when system is running.
 - Location - C:\WINNT\SYSTEM32\CONFIG
- SYSKEY
 - Utility which moves the encryption key for the SAM database off of the computer

Passwords

Impact on Security

- Simple hacking tools are available to anyone who looks for them on the Internet.
- Tools such as **LOphtCrack** allow admittance into almost anyone's account if a simple eight-digit password is used.

People are frightened when they learn that using only an eight-digit password with standard numbers and letters will allow anyone to figure out their passwords in less than two minutes when one downloads a publicly available tool like LOphtCrack from the Internet. This was the kind of tool which we found (in Al Qaeda's arsenal), nothing terribly sophisticated.

- Richard Clark, Presidents Advisor on Cyber Security (2001-2003)

- Sometimes even hacking tools aren't even necessary

Passwords

Threats to Password Security, Part 1

- Disclosure
 - Voluntary disclosure of information
 - Inadequate guarding of system passwords
- Inference
 - Known pattern to creation of passwords
 - Use of generated passwords with predictable algorithm
- Exposure
 - Accidental release of password
- Loss
 - Forgetting to remember passwords
 - Can lead to creation of easy passwords

Passwords

Threats to Password Security, Part 2

- Snooping/Eavesdropping
 - Keyloggers
 - Network sniffing (intercepting of network communication where a password is submitted)
- Guessing
 - Limited amount of choices which can be figured out through process of elimination
 - Use of blank/common passwords, passwords which can be figured out by knowing name of relatives, pets, etc.
- Cracking
 - Automated “guessing”

Passwords

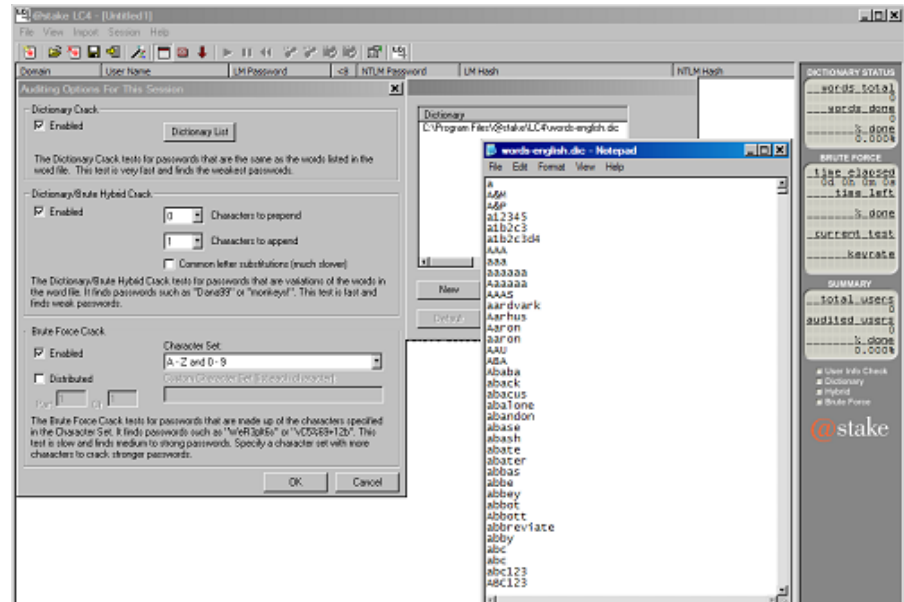
Why Cracking is Possible

- Passwords are NOT truly random
 - 52 upper/lowercase letters, 10 digits, and 32 punctuation symbols equals ≈ 6 quadrillion possible 8-character passwords
 - People like to use dictionary words, relative and pet names equaling ≈ 1 million common passwords
 - On average, each person has 8-12 passwords:
 - Different systems impose different password requirements.
 - Passwords need to be changed often.
 - Some passwords are only used occasionally.

Passwords

Dictionary Attack

- Attacker can compute $H(\text{word})$ for every word in a dictionary and see if the result is in the password file
- With 1,000,000-word dictionary and assuming 10 guesses per second, brute-force online attack takes 50,000 seconds (14 hours) on average
 - This is very conservative; Offline attack is much faster!



Passwords

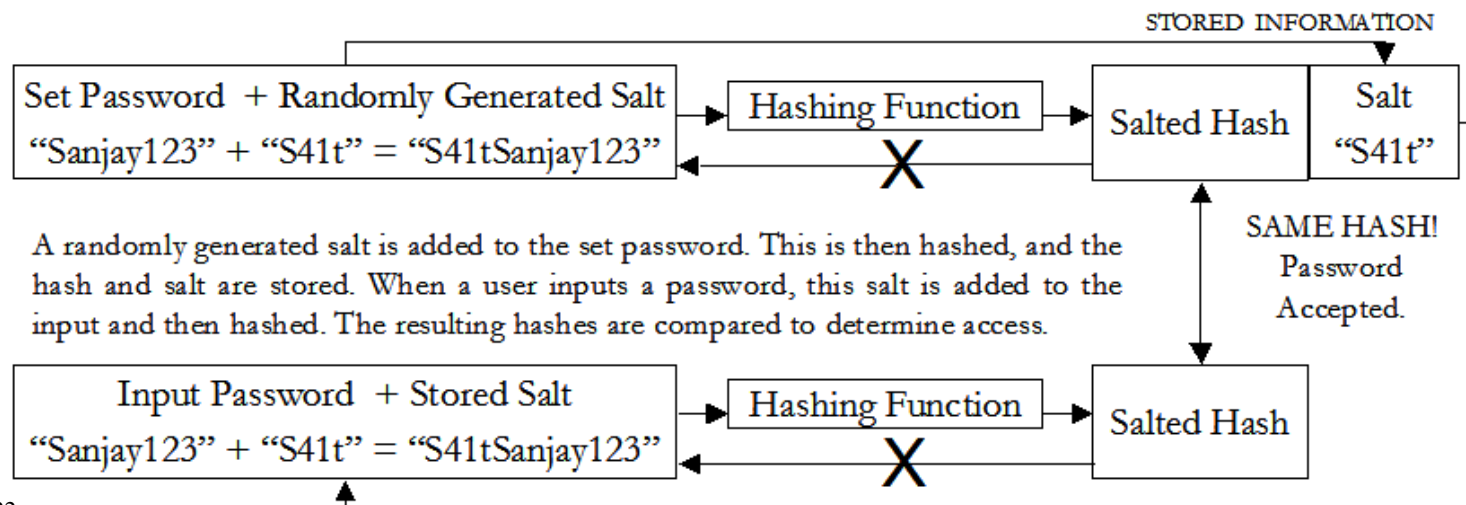
Types of Password Cracking

- Dictionary Attack
 - Quick technique that tries every word in a specific dictionary
- Hybrid Attack
 - Adds numbers or symbols to the end of a word
- Brute Force Attack
 - Tries all combinations of letters, numbers & symbols
- Popular programs for Windows password cracking
 - LophCrack (discontinued by Symantec when acquired @stake)
 - Cain & Abel (UNIX)
 - John the Ripper (UNIX)
 - Sam Inside

Passwords

Cracking Protection - Salting

- Salting requires adding a random piece of data and to the password before hashing it.
 - This means that the same string will hash to different values at different times
 - Users with same password have different entries in the password file
 - Salt is stored with the other data as a complete hash
- Hacker has to get the salt add it to each possible word and then rehash the data prior to comparing with the stored password.



Passwords

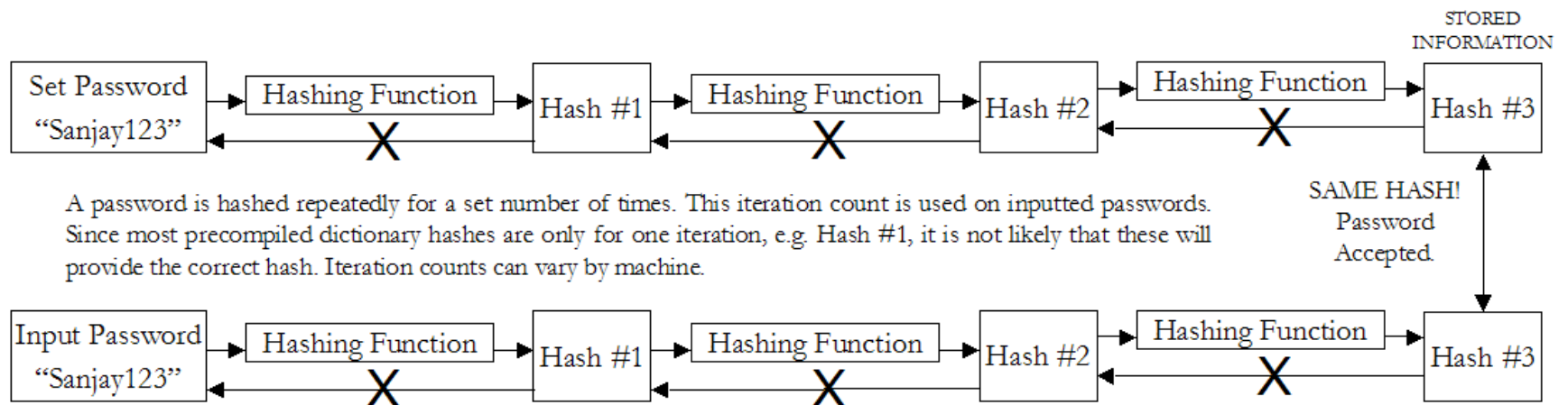
Cracking Protection - Salting Advantages

- Without salt, attacker can precompute hashes of all dictionary words once for all password entries
 - Same hash function on all UNIX machines
 - Identical passwords hash to identical values; one table of hash values can be used for all password files
- With salt, attacker must compute hashes of all dictionary words once for each password entry
 - With 12-bit random salt, same password can hash to 212 different hash values
 - Attacker must try all dictionary words for each salt value in the password file

Passwords

Cracking Protection - Iteration Count

- The same password can be rehashed many times over to make it more difficult for the hacker to crack the password.
- This means that the precompiled dictionary hashes are not useful since the iteration count is different for different systems
 - **Dictionary attack is still possible!**



Passwords

Authentication Protocols

- TRANSFORMED PASSWORD
 - Password transformed using one way function before transmission
 - Prevents eavesdropping but not replay
- CHALLENGE-RESPONSE
 - Server sends a random value (challenge) to the client along with the authentication request. This must be included in the response
 - Protects against replay
- TIME STAMP
 - The authentication from the client to server must have time-stamp embedded
 - Server checks if the time is reasonable
 - Protects against replay
 - Depends on synchronization of clocks on computers
- ONE-TIME PASSWORD
 - New password obtained by passing user-password through one-way function n times which keeps incrementing
 - Protects against replay as well as eavesdropping

Passwords

Challenge Response

- User and system share a secret key
- Challenge: system presents user with some string
- Response: user computes response based on secret key and challenge
 - Secrecy: difficult to recover key from response
 - One-way hashing or symmetric encryption work well
 - Freshness: if challenge is fresh and unpredictable, attacker on the network cannot replay an old response
 - For example, use a fresh random number for each challenge
- Good for systems with pre-installed secret keys
 - Car keys; military friend-or-foe identification

Passwords

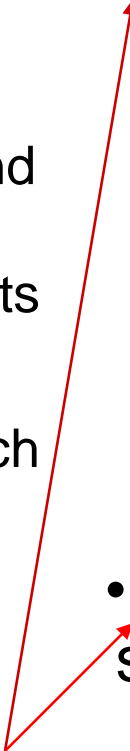
Personal Token Authentication

- Personal Tokens are hardware devices that generate unique strings that are usually used in conjunction with passwords for authentication
 - A variety of different physical forms of tokens exist
 - e.g. hand-held devices, Smart Cards, PCMCIA cards, USB tokens
 - Different types of tokens exist:
- **Storage Token:** A secret value that is stored on a token and is available after the token has been unlocked using a PIN
 - **Synchronous One-time Password Generator:** Generate a new password periodically (e.g. each minute) based on time and a secret code stored in the token
 - **Challenge-response:** Token computes a number based on a challenge value sent by the server
 - **Digital Signature Token:** Contains the digital signature private key and computes a digital signature on a supplied data value



Passwords

Improving Security

- Password complexity
 - Case-sensitivity
 - Use of special characters, numbers, and both upper and lower-case letters
 - Minimum length requirements
 - Security questions
 - Ask personal questions which need to be verified
 - Some questions are very easy to discover answers
 - Virtual keyboard
 - Person clicks on-screen keyboard to enter password (prevents keylogging)
 - Single sign-on
 - User only has to remember one password at a time and yet can access all/most of their resources
 - AKA Enterprise Reduced Sign-On (almost impossible to have one password used for everything due to integration issues)
 - Centralized password storage management
 - Online sites accessible through one password which contain all other passwords
- Single point of failure, but easier to remember
- 

Passwords

Improving Security

- Graphical passwords
 - Goal: increase the size of memorable password space
- Rely on the difficulty of computer vision
 - Face recognition is easy for humans, harder for machines
 - Present user with a sequence of faces, he must pick the right face several times in a row to log in
- Other examples
 - Click on a series of pictures in order
 - Drawing a picture
 - Clicking four correct points on a picture
- Reading graphical text
 - Requires user to input text based on what is seen in the graphic. Attempts to curb automated password crackers due to difficulty in distinguishing letters/numbers
 - Scheme where users had to input text based on graphics shown to “undress” a picture



Passwords

Biometric/Behaviometric Authentication

- Uses certain biological or behavioral characteristics for authentication
 - Biometric reader measures physiological indicia and compares them to specified values
 - It is not capable of securing information over the network
- Biological Examples
 - Fingerprint, Iris, Retina, Face, & Hand Recognition
- Behavioral Examples
 - Handwriting, Gait, Typing Rhythm, Mouse Gesture Recognition

Passwords

Biometric Considerations

Universality	How commonly biometric is found
Uniqueness	How well biometric distinguishes between others
Permanence	How well biometric resists aging
Collectability	How easy biometric is to acquire
Performance	Accuracy, speed, and robustness of system capturing biometric
Acceptability	Degree of approval by the public for use
Circumvention	How hard it is to fool authentication system

Passwords

Protection/Detection

Protection:

- Disable storage of LAN Manager hashes.
- Configure both Local and Domain Account Policies (Password & Account Lockout Policies).
- Audit access to important files.
- Implement SYSKEY security on all systems.
- Set BIOS to boot first from the hard drive.
- Password-protect the BIOS.
- Enforce strong passwords!
- Change your passwords frequently.
- Use two or three factor authentication.
- Use one time passwords.

Passwords

Ten Common Mistakes

1. Leaving passwords blank or unchanged from default value.
2. Using the letters p-a-s-s-w-o-r-d as the password.
3. Using a favorite movie star name as the password.
4. Using a spouse's name as the password.
5. Using the same password for everything.
6. Writing passwords on post-it notes.
7. Pasting a list of passwords under the keyboard.
8. Storing all passwords in an Excel spreadsheet on a PDA or inserting passwords into a rolodex.
9. Writing all passwords in a personal diary/notebook.
10. Giving the password to someone who claims to be the system administrator.