Authentication – binding of identity to a subject

External entity must provide information to enable system to confirm its identity.

Information comes from one or more of the following:

What the entity knows – passwords, secret information

What the entity has - badge, card

What the entity is – fingerprint, retinal characteristics

Where the entity is – in front of a terminal

Authentication process consists of obtaining authentication information from an entity, analysing data and determining if it is associated with the entity.

Secure system – system of secure states.

Subjects and objects within a secure system.

They interact according to the access control matrix.

This also states and state transitions. Keeping within a bubble.

Secure system is secure so long as I don’t move , outside of it.

Passwords – authentication mechanism based on what people know.

User supplies a password and computer validates it. If correct authentication succeeds. If wrong authentication fails.

Password – information associated with an entity that confirms the entity’s identity

Password space – set of all sequences of characters that can be passwords.

Password selection

Goal of password selection – Make difficulty of guessing password as great as possible without compromising ability of user to use password.

Random selection of password

Passwords are generated randomly from some set of characters and some set of lengths. This minimizes chances of attacker guessing a password.

Let expected time to guess password -T. T is maximum when selection of any of a set of passwords – equiprobable.

Compromise between using random unmemorizable passwords and writing passwords down – use pronounceable passwords.

User selection of passwords

User created passwords more memorable than computer generated passwords.Countering this advantage, users tend to select familiar passwords such as dictionary words.

When users select password, selection mechanism should constrain what passwords users are able to select.

This technique – proactive password selection enables users to propose passwords they can remember but rejects any deemed too easy to guess.

Categories of passwords easy to guess

Account names

Usernames

Computer names

Dictionary words

Patterns from keyboard

Passwords used in the past

Short Passwords easily found.

Proactive password checker – software enforcing specific restrictions on selection of new passwords.

These restrictions – password policy

Extension of password – passphrase

Passphrase – password composed of multiple words, possibly other

characters.

Given advances in computing power, passwords once deemed secure now easily discoverable.

Passphrase is password composed of multiple words and possibly other characters.

Passphrase increases length of passwords while allowing user to pick something easy to remember.

Widely used method for keeping track of passwords – encipher them and keep them in a password wallet.

Password wallet or password manager – mechanism that encrypts a set of user’s passwords.

Wallet allows users to store multiple passwords in a repository itself encrypted with a single cryptographic key, so users only need to remember that key.

2 issues -accessibility, cascading disclosure

Master key – quite complex and is only password user needs to remember.

If master password discovered, all passwords are disclosed.

Attacking passwords

Dictionary attack – act of guessing password by repeated trial and error. Name of attack comes from list of words (a dictionary) , used for guesses.

Dictionary – set of commonly used passwords

A set of strings in decreasing order of probability of selection.

Salting

If offline dictionary attack aimed at finding any user’s password , technique of salting increases amount of work required.

Salting makes choice of complementation function a function of randomly selected data.

a "salt" in authentication is a random data that is used as an additional input to a one-way function that hashes a password or passphrase. Its primary purpose is to defend against dictionary attacks and pre-computed rainbow table attacks

1. **Randomness**: When a user creates or updates their password, the system generates a random salt. This salt is unique for each user or even each password change.
2. **Hashing with Salt**: The salt is combined with the password and then hashed. This means the actual password is never stored in the database; instead, the hashed version of the salted password is stored.
3. **Unique Hashes**: Because each user's password is salted with a different random value, even if two users have the same password, their stored password hashes will be different. This uniqueness helps prevent attackers from using pre-computed tables (rainbow tables) to reverse-engineer passwords.

Why are passwords the most common authentication method ?

We use it a lot.

Passwords have a lot of problems.

They are an authentication method that are subject to being cracked.

1)Brute force

Difficult to do brute force through keyboard.

There are other situations where a brute force attack is viable.

Might have a network authentication where a saved password is used.

How are passwords represented on the system.

Hash functions are one way functions

If we have a file with hash values.

Can hash many passwords and see if they match hash value.

If system don’t have salt it is vulnerable to rainbow tables attack.

Salting  
a "salt" in authentication is a random data that is used as an additional input to a one-way function that hashes a password or passphrase. Its primary purpose is to defend against dictionary attacks and pre-computed rainbow table attacks.

Here's how salt works in authentication:

1. **Randomness**: When a user creates or updates their password, the system generates a random salt. This salt is unique for each user or even each password change.
2. **Hashing with Salt**: The salt is combined with the password and then hashed. This means the actual password is never stored in the database; instead, the hashed version of the salted password is stored.
3. **Unique Hashes**: Because each user's password is salted with a different random value, even if two users have the same password, their stored password hashes will be different. This uniqueness helps prevent attackers from using pre-computed tables (rainbow tables) to reverse-engineer passwords.
4. **Verification**: When a user logs in, the system retrieves the salt associated with that user, combines it with the entered password, and hashes this combination. If the resulting hash matches the stored hash, the password is considered correct.
5. **Enhanced Security**: Salting makes it significantly more difficult for attackers to use brute-force methods to crack passwords because they must compute hashes for each possible password combined with each possible salt.

In summary, salt in authentication adds an extra layer of security to password storage and verification processes, making it more challenging for attackers to crack passwords through common methods like dictionary or rainbow table attacks.

Rainbow tables attack

A rainbow table in the context of computer security is a precomputed table used for reversing cryptographic hash functions, primarily for cracking password hashes. Hash functions are used in various security applications to transform passwords into a string of characters (hash), which is stored instead of the actual password. Rainbow tables are designed to attack systems that use unsalted hashes.

Here's a breakdown of how rainbow tables work and their role in authentication:

1. **Hash Functions**: When a password is entered, it's transformed by a hash function into a fixed-size string of characters, which is difficult to reverse. This hashed version is stored instead of the plaintext password.
2. **Precomputed Hashes**: A rainbow table contains a large precomputed list of plaintext passwords and their corresponding hash values. These tables are generated by applying hash functions to a range of potential plaintext passwords.
3. **Efficiency**: Rainbow tables use a method of storing hashes that balances the size of the table with the speed of lookup. This makes them more efficient in terms of storage than simple hash tables but slower in lookups than more straightforward hash chaining methods.
4. **Cracking Process**: To crack a password, the attacker searches the rainbow table for the stored hash. If a match is found, the table provides the corresponding plaintext password.
5. **Limitations and Countermeasures**:
   * **Salting**: To defend against rainbow table attacks, many systems add a unique random value, known as a salt, to each password before hashing. This means the same password will have a different hash for each user or each time it's changed, rendering rainbow tables ineffective.
   * **Storage and Computation**: Rainbow tables are practical for simpler and shorter passwords. However, as password complexity increases (in length and character variety), the size of the rainbow table needed to cover all possible combinations becomes impractical.
   * **Modern Hashing Algorithms**: More advanced hashing algorithms include features that make the use of rainbow tables inefficient or impractical.

In summary, rainbow tables are a tool for cracking passwords by using precomputed hash values, but their effectiveness is greatly reduced by modern security practices like using salted hashes and complex hashing algorithms.

Build a big database of hashes and the passwords, then can just lookup.

2)Try likely/probable

Took information from guys account, working from that they found a key that was likely a password that was likely.

Understanding things about individual will help to search through space of passwords.

Words more likely than random letters

Dictionary attack will try with most likely combinations of letters first.

Dictionary attack has its own dictionary ,a dictionary made up of common passwords

qwerty there in dictionary not there in normal dictionary

12345

Good dictionary attack will also try combinations around things it does know.

Alan01

As time goes, dictionary attack will run out of common words or combinations. Eventually degrades to brute force attack.

3)Look for them

Humans write passwords down.

4)Ask the user

Social engineering – art of convincing someone to do something

5)Eavesdropping

Listening to keystrokes on keyboard.

Breaking into router to observe network traffic.

How to make and keep good passwords

Lists with advice in them.

Analyse list with good advice in terms of things we already understand.

1)use characters other than a-z

If no of letters – 8 ,then space is 26^8.  
A-Z 52^8

2)choose long passwords

86^10 , play game and raise power, raise base

Avoid actual names or words.

Good technique –

* Use characters other than just a-z
* Choose long passwords
* Avoid actual name or words
* Use a string you can remember
* Use variants for multiple passwords
* Change the password regularly
* Don't write it down
* Don't tell anyone else

Help in face of too many passwords

Mechanisms to help save passwords

Password rememberers – Browser pages that save passwords

Key rings – Not done so much by the browser, but built into the OS that helps to handle keys.

Password put onto a data structure. This data structure is then encrypted securely once you log out .Preferably they only decrypted for a minimum amount of time.

Users have a problem with passwords, they need help

Come to a new system that can pull out password.

Keyrings, in the context of computer security and password management, are secure digital vaults used to store and manage passwords and other sensitive information like cryptographic keys. The concept of a keyring helps users maintain a collection of passwords and other credentials securely, without the need to remember each one individually. Here's an overview of how keyrings work

1. **Secure Storage**: Keyrings provide encrypted storage for passwords and keys. This means that the stored information is protected by strong encryption, making it inaccessible to anyone without the proper credentials to unlock the keyring.
2. **Master Password**: Typically, a keyring is secured with a single master password. This is the only password a user needs to remember. Once the master password is entered, the user can access all the stored credentials in the keyring.
3. **Integration with Applications**: Keyrings are often integrated with software applications, such as web browsers and email clients, to manage passwords for various accounts. For example, a browser might use a keyring to store and autofill passwords for different websites.
4. **Convenience and Security**: By using a keyring, users can maintain strong, unique passwords for each of their accounts without the need to memorize each one. This practice enhances security, as it encourages the use of complex passwords that are less susceptible to being guessed or cracked.
5. **Types of Keyrings**: Keyrings can be found in various forms, such as:
   * **Local Keyrings**: Stored locally on a user's device, often used in operating systems like Linux and macOS.
   * **Cloud-based Password Managers**: Services that store your passwords securely in the cloud, allowing access from multiple devices. Examples include LastPass, Dashlane, and 1Password.
6. **Security Features**: In addition to encryption, many keyring services offer additional security features, such as two-factor authentication (2FA), secure password generation, and the ability to audit password strength.
7. **Backup and Recovery**: Keyrings often provide options for securely backing up and recovering stored data, which is crucial in case of device loss or failure.

In summary, keyrings are a secure and convenient way to manage passwords and other sensitive credentials. They reduce the burden of remembering multiple complex passwords, thereby contributing significantly to both user convenience and overall cybersecurity.

Your answer is partially correct.

You have correctly selected 3.

The correct answer is: A bank card PIN code → Eavesdropping, A short but random combination of characters → Brute force attack, A Lithuanian swearword → Dictionary Attack, A password transported in clear → Sniffing Attack

Online dictionary attacks

Attacker supplies guess g to authentication function, if it returns true

g is correct password.

In the context of cybersecurity, dictionary attacks are a method used by attackers to crack passwords. The difference between online and offline dictionary attacks primarily lies in how and where they are conducted:

1. **Online Dictionary Attacks**:
   * **Location**: These attacks are carried out directly against a live system or service (e.g., a website, a network login portal).
   * **Method**: The attacker tries to log in using different passwords from a pre-compiled list (the 'dictionary'), which often contains common or previously leaked passwords.
   * **Detection and Prevention**: Online attacks can be detected relatively easily by monitoring for multiple failed login attempts. Services can implement disabling – n attempts to login to account fail, account is disabled.
   * **Speed**: These attacks are slower due to network latency and the risk of detection, which forces attackers to try fewer passwords over a longer period.
   * **Risk to Users**: Users are at risk as long as the attack is ongoing, and they may experience inconvenience due to account lockouts or additional security measures.

1. **Offline Dictionary Attacks**:
   * **Location**: These attacks are carried out on a copy of the target's password database that the attacker has somehow obtained.
   * **Method**: The attacker uses the dictionary to try and match the hashed passwords in the database. Advanced methods may include using rainbow tables or hash tables to speed up the process.
   * **Detection and Prevention**: Offline attacks are much harder to detect since they don’t interact with the live system. Prevention relies on securing the password databases and using robust hashing algorithms with salts.
   * **Speed**: Offline attacks can be much faster as they are limited only by the attacker's hardware capabilities and don't have the same risks of detection and account lockout.
   * **Risk to Users**: Users may not be immediately affected, but if the attack is successful, their credentials can be compromised without their knowledge, leading to potential unauthorized access later.

In summary, online dictionary attacks are direct attacks against a live system with slower execution and easier detection, while offline dictionary attacks are faster and stealthier, targeting a stolen copy of the password database. Both types of attacks underscore the importance of using strong, unique passwords and implementing robust security measures.

Password strength

How well password selection schemes work to produce passwords that are difficult to guess requires examination of selected passwords.

Guessing entropy – amount of work to guess password for user.

Password aging

Requirement that password be changed after some period of time passed or some event occurred.

Expected time to guess password – 90 days.

Changing password more frequently than 90 days, reduce prob that attacker can guess password still being used.

Aging by itself ensures little as estimated time to guess password is average.

If password easy ,time to guess password less than average.

It should be used with other methods.

1. **Use Characters Other Than Just a-z**: This refers to including a mix of uppercase and lowercase letters, numbers, and special characters. Such a mix increases the complexity of the password, making it harder for attackers to guess or crack.
2. **Choose Long Passwords**: Longer passwords are generally more secure because they increase the number of possible combinations that an attacker must try to crack the password.
3. **Avoid Actual Names or Words**: Using common names, dictionary words, or easily guessable information (like birthdates or anniversaries) makes passwords more vulnerable to dictionary attacks.
4. **Use a String You Can Remember**: This is crucial for avoiding the need to write down your password. A memorable yet complex password can be created by using a passphrase, a combination of unrelated words, or a mnemonic device.
5. **Use Variants for Multiple Passwords**: Using slightly different passwords for different accounts can help limit the damage if one of your passwords is compromised. However, the variations should still be significant enough to ensure each password is unique.
6. **Don't Write It Down**: Writing down passwords can be a security risk, especially if the note is easily accessible. If remembering multiple complex passwords is difficult, consider using a reputable password manager.
7. **Don't Tell Anyone Else**: Keeping your passwords confidential is fundamental. Sharing passwords increases the risk of them being misused or accidentally disclosed.

Password aging useless if user can change password to same thing.

Technique to prevent this – record n previous passwords.