10 security principles:

1) Economy of mechanism : Keeping security measures simple → effective

2) Fail-safe defaults:
keep thing safe, starting with strict rules
→ new users ~~don't~~ have minimal access rights to services

Final 3) Complete mediation
check permission access to resources every time when requested.
Ex: → bkash payment needs PIN to prevent unauthorized access

4) Open design: security design and architecture → public
cryptographic key → private
Pros: others can find and fix issues

5) Separation of privilege
→ multiple conditions should be fulfilled to achieve access to resources
→ 2 instead of 1 keys

6) Least privilege: program/ user get the minimum privilege that's necessary to complete the job.

7) Least common mechanism: → limit sharing
when multiple users access to a file
→ need separate channel to access
→ one's action doesn't affect other's security

8) Psychological acceptability: clean interface
fulfill expected security

9) ~~Work factor:~~ compromise recording:
not focus on stopping intruder
not using complex ways to stop them
– just watch and learn

10) ~~Compromise recording~~: Work Factor
– think how much it cost someone to break security vs. what they've.
– students grade < millitary info
↓
need more secury

A) Access control has 4 mechanisms:
1) Access control metrices: A table that sets rules for who can do what with certain things.

col — file, folden each cell — what can do

	/etc/pass	usr/bin
root	R, W	R, W, exec
mike	R	R exec

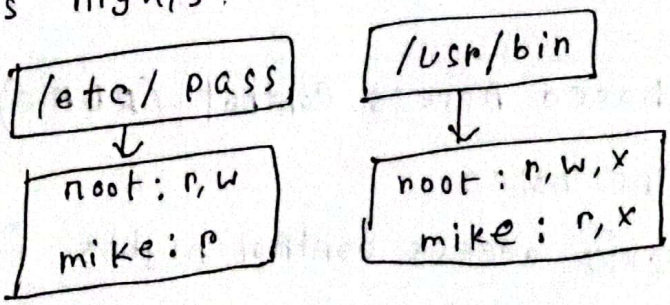
now → person group system

Pros: 1) easily see a cell and and tell which user can do what with a file

2) easy format

Cons: huge size, complex, hard to manage, time-consuming

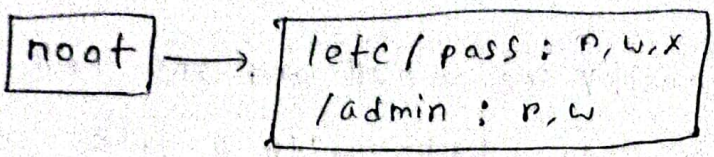
2) Access control lists: Each object/file has a list of users who can access it and their access rights.



Pros: smaller, convinent to use

Cons: Finding out all the access rights for a user → slow

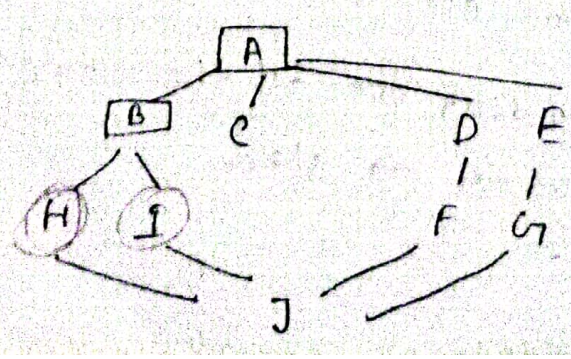
3) **Capabilities**: For each user, it lists the objects they can access and what they can do with each object.



Pros :- Admins only need to handle access for pairs with actual rights
- reduce workload
- easy to find

Cons -complex,
- slow, because rights for a file are not directly linked

4) **Role based Access Control (RBAC)**
→ define roles
→ specify access control rights for these roles
→ not for subjects directly



B can inherit the access rights of H, I

Pros: 1) have fewer roles than subjects
2) easy to manage access rights
3) easy check to see if any user has rights

Cons:
1) not commonly used in current OS

final * Describe a process through which a system can fulfill the complete mediation security principal.

→ repeat for every attempt
1) when user access an object, they send a request to the system.
2) check authorization.
3) verify permission.

- Cost of circumventing: resource needed for attacker to defeat a security system
- Security by obscurity: keep algorithms private → opposite of open design
- Kerckhoff's principle

→ A cryptosystem should be secure even if everything about the system, except the key, is public knowledge