Authorization: is the concept of allowing access to resources only to those permitted to use them.

1) CPU operations:-
   Consist of 2 modes:-
   a) User mode - has restrictions
   b) Supervisor mode - must run trusted code

2) Memory:
   Memory is protected by avoiding unauthorized reading/writing of memory.
   This can be achieved as follows:-
   a) Only mapping pages that process has permission to access.
   b) Clear pages when we recycle them.
No devices

- Users processes access these through the OS.

Examples: Dish, Network.

3 Dimensional View of Authorization

Permissions:
Given an Object: What users can perform what operations?

Capabilities:
Given a user: What operations can be performed on what objects?
File systems have methods of administering permissions to:
- users
- group of users (world)

Unix

Three specific permissions on Unix-like systems are:
- read
- write
- execute

Example:

File: owner.

owner world
rwX rwX.

File: owner, group.

owner group world
-w-r--r--.
ACL's: Access Control Lists

ACL is a list of permissions attached to an object. This list specifies who or what is allowed to access the object and what operations are allowed to be performed on the object.

user/group : permission : allow/deny

Objects
Files
Processes

Operations
read / read directory
modify
delete
execute
create
kill

Example:

a/b/c
If a, b are protected then one cannot go to c.

Alternatively, to deny access to a directory on objects is to have capabilities for user (or process) what ops on which dirs?
PRINCIPLE OF LEAST PRIVILEGE

Each part of the system should have privilege to do things but not additional once.

Example: Application like Blackboard running on single computer.
Create set of files that can be read/written through particular program.

setuid in Unix

→ new userid : gradebk
  data : user gradebook
  amx----

grade app : user gradable
  setuid -x -x -x

When u run if, it changes uid to or met itself.

[SE LINUX]

Security Enhanced Linux

PTO.
→ Separation of Mechanism & Policy.

→ You can write your own policy.
Network Attacks

Buffer Overflow attack.

Char buf [128];
gets (buf);

Stack frame:

```
ret
```

```
addr
```

```
buf[]
```

```
ADD 20, sp
:  
sub 20, sp
ret
```

```
000
A000
```
```
A010
bad code
```
```
4050
```
Defense against Buffer overflow:

1) dont hr them
2) NX or do not execute bit

Add execute permission bit to execute then you can prevent attacks.

3) ASLR - Addr space layout randomization.