COMPUTER SYSTEMS

LECTURE NOTES: PART II

Authorization:

- lowest level we have CPU ops. where we have user/supervisor mode.
- User → have some restrictions
- Supervisor → can do anything.
- must run trusted code.

Memory:
- avoid unauthorized reading/writing of memory.
- we do this by only mapping pages that process has permission to access.
- We have to clear pages when we recycle them.

I/O Devices:
- disk, network.

User processes access data through OS.

Dimensions:
- operations
- objects
- users
* Permissions:
  Given object:
  what users can perform what operations?

* Capabilities:
  Given a user:
  what ops can be performed on what objects?

Each file has an owner,
we have 3 sets of permissions:
  0 owner 10 world
UNIX has rwx permissions

File has owner and group,
& each user also has group.

owner group world
rwx rwx r-x

eg: UsedID: Pjd
    Member of group: faculty
    CSG112

  pjd can access files & folders from
  both the groups.
ACLs: Access control list

Using Right-click -> properties -> security

We can specify accesses on files in Windows.

usu/group : permission: allow/deny

<table>
<thead>
<tr>
<th>Objects</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Files</td>
<td>Read / readdir</td>
</tr>
<tr>
<td>devices</td>
<td>Modify</td>
</tr>
<tr>
<td>processes</td>
<td>delete</td>
</tr>
<tr>
<td></td>
<td>Create</td>
</tr>
<tr>
<td></td>
<td>execute</td>
</tr>
<tr>
<td></td>
<td>Kill</td>
</tr>
</tbody>
</table>

e.g. la/b/c.txt

to get to c.txt we need to have permissions to directory a & b.

UNIX basically uses rwx.

For kill, you can kill only your processes.

Capabilities:

for user (or process)

What ops on which objs?
In Windows, we have local login, net login, service login.
We also have load driver, shutdown.

In Linux, we have chown, kill process load module (device) of other user.

principle of least privilege:
Each part of the system should have privileges to perform ops it needs to do.

```
data  app
```

& blackboard user login
to check for permissions.

- new uid: gradebk
  data : user gradebk
  setuid, setgid

  gradeapp: user gradebk
  setuid, - - - -
In the next lecture we'll cover security enhanced Linux.

Processes

SE Linux

Configuration

Objects

Operations

Separation of mechanism & policy.

Attacks:

Buffer overflow attack:
Its usually a new attack.

char buf[128];
gets(buf);

In our case, local vars will be 128 bytes long.

F: ADD 20, SP
SUB 20, SP
RET
If we provide more than 128 bytes of data then we overwrite the return address.

Defense against buffer overflow:
1. don't have them.
2. NX or do not execute bit
3. ASLR: address space layout randomization