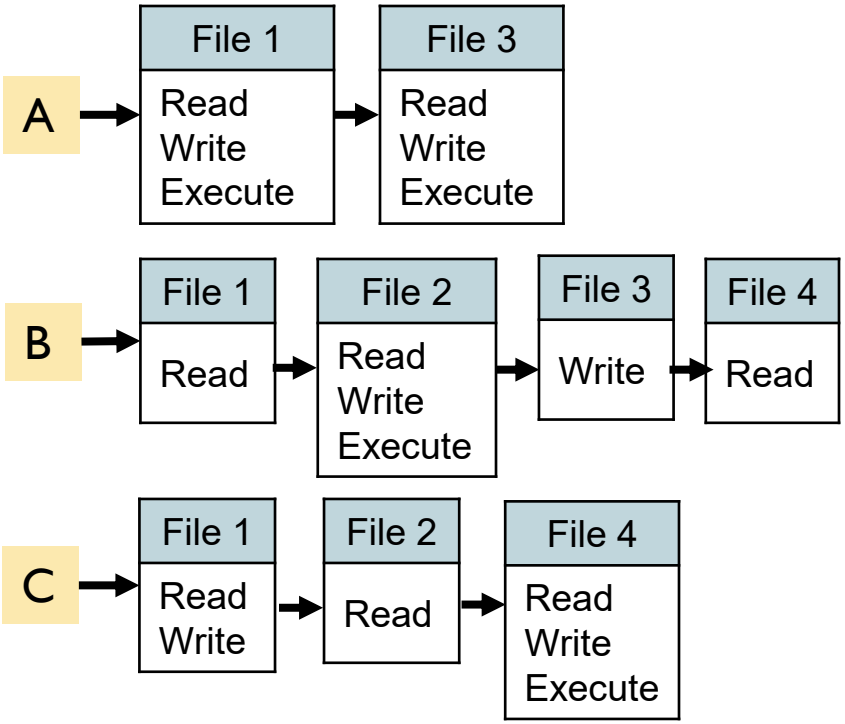
Capability List (C-List)

In practice, an access control matrix is usually sparse and can be implemented by decomposition in one of two ways

Decomposition by rows

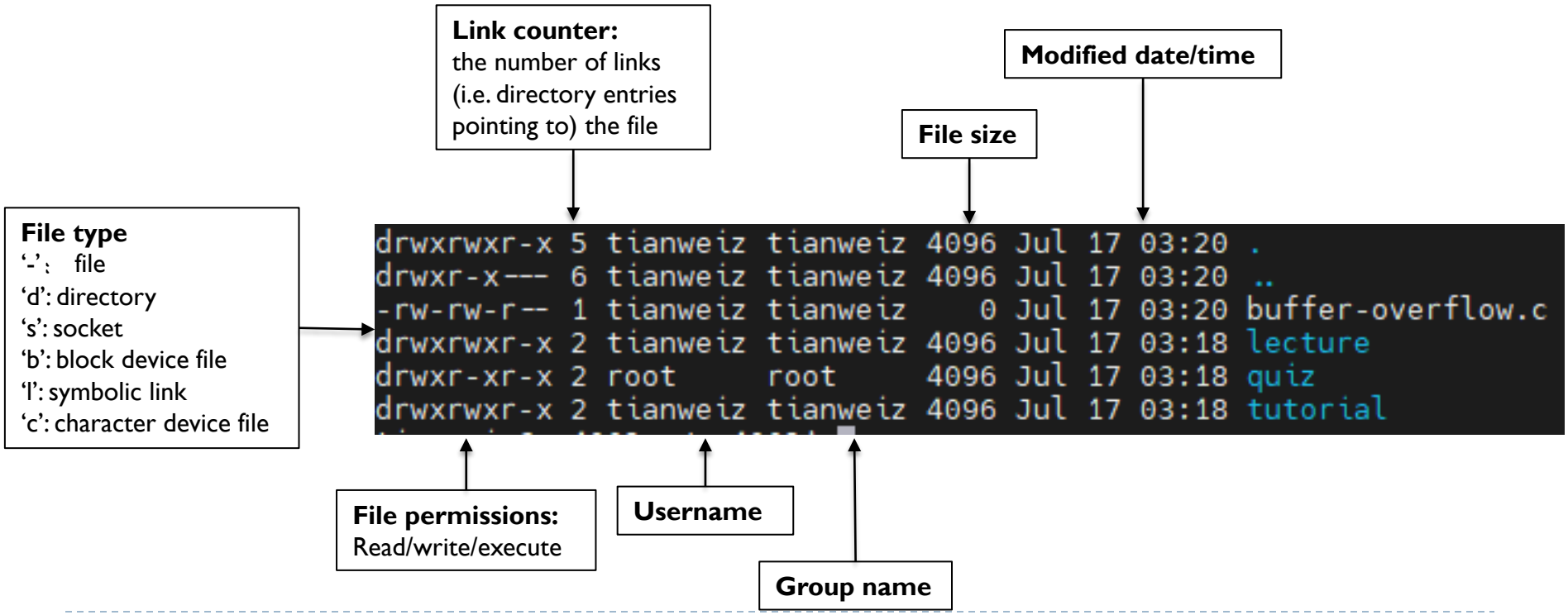
- C-list specifies authorized objects and operations for a particular user.
- C-List is convenient when determining the access rights available to a specific user.



Example: Resource Management in Unix OS

Files, directories, memory devices, I/O devices are uniformly treated as resources

- These resources are the objects of access control.
- Each resource has a single user owner and group owner



Permission Representation

Three permissions with three subjects

- Read, Write, Execute
- Owner, Group, Other
- Examples:
  - rw-r--r--: read and write access for owner, read access for group and other.
  - rwx------: read, write, and execute access for owner, no rights to group and other.

Octal Representation

- rw-r--r--: 110 100 100: 644
- rwx------: 111 000 000: 700

Adjust permission:

- Users can change the permissions:
  - chmod 754 filename
  - chmod u+wrx,g+rx,g-w,o+r,o-wx filename
- root can change the ownerships:
  - chown user:group filename

Controlled Invocation

Superuser privilege is required to execute certain OS functions

- Example: password changing
  - User passwords are stored in the file /etc/shadow
  - This file is owned by the root superuser. A normal user has no access to it
  - When a normal user wants to change his password with the program passwd, this program needs to give him additional permissions to write to /etc/shadow

SUID: a special permission flag for a program

- If SUID is enabled, then user who executes this progam will inherit the permissions of the program's owner.
- A normal user executing passwd can get additional root permission to write the new password to /etc/shadow

The execute permission of the owner is given as s instead of x

```
root@cx4062:~# ls -al /usr/bin/passwd
-rwsr-xr-x 1 root root 59976 Mar 14 08:59 /usr/bin/passwd
```

Security of Controlled Invocation

Many other SUID programs with the owner of root

- /bin/login: login; /bin/at: batch job submission; /bin/su: change UID

Potential dangers

- As the user has the program owner's privileges when running a SUID program, the program should only do what the owner intended
- By tricking a SUID program owned by root to do unintended things, an attacker can act as the root

Security consideration

- All user input (including command line arguments and environment variables) must be processed with extreme care
- Programs should have SUID status only if it is really necessary.
- The integrity of SUID programs must be monitored.

Logging, Monitoring & Auditing

Purposes

- Intrusion detection: identify unauthorized access or system changes.
- Forensics and investigation: provide historical data for incident response.
- Accountability: track user actions and commands.
- Performance monitoring: assist in debugging applications and diagnosing.

Challenges

- High storage and processing requirements: precisely select and record the most critical data.
- Attackers may erase or modify logs: well protect the data, e.g., via encryption and access control.
- May compromise user privacy: follow the compliance and retention policies.

Examples of Monitored Data

The OS collects different types of data at different layers.

- System call traces: describe the activities or behaviors of processes running in the system.
- Log file: information on user activity, including user' login record, history of commands, etc.
- File integrity checksums: periodically scan critical files for changes and compare cryptographic checksums for these files, with a record of known good values.
- Registry access: monitor access to the registry. This is specific to Windows operating systems.
- Kernel and driver-level monitoring: this source provides insight into OS kernel-level anomalies.
- Resource usage: CPU, memory or I/O utilization and activities can indicate the execution of some malicious behaviors.
- Network activities: include established connections and received packets

Intrusion Detection

Intrusion Detection System (IDS)

- A system used to detect unauthorized intrusions into computer systems.
- IDS can be implemented at different layers, including network-based IDS, host-based IDS.
- We mainly focus on host-based IDS, which monitors the characteristics of a single host for suspicious activities.

An IDS comprises three logical components:

- Sensors: responsible for collecting data.
- Analyzers: responsible for determining if an intrusion has occurred, and the possible evidence. It may provide guidance about what actions to take as a result of the intrusion.
- User interface: enables a user to view output from the system or control the behavior of the system.