Feb 21, 2005 -- Lecture 14

22C:169
Computer Security

Douglas W. Jones

Department of Computer Science

File Protection

File system protection

First file systems,

no protection

Ownership-based protection

owner write access, others read-only

Access control lists

Fully general access control

Unix

A step backward

Ownership-based access control

Each file has an owner

Typically, designated by a user ID

Rights granted depend on user of file

If user is owner, all rights granted

If user is not owner, limited rights

Inflexible

Not always natural

Generalization

Access rights are variable

eg: read-write, read-only, no access

Only the owner may change rights of file

rights applying to owner

rights applying to others

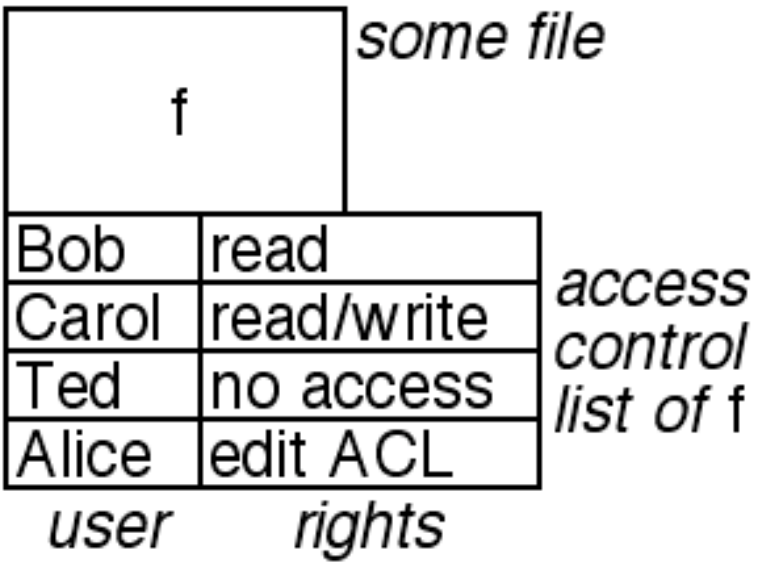
Others limited to what owner permits

Owner not always trustworthy

Ownership insufficient

Access Control Lists

Multics File System, Daley and Neumann, AFIPS 1965



Multics File System

File system was hierarchical

`users>jones>class>file`

File ACLs had rights

read, write, edit ACL ...

Directory ACLs had rights

traverse, read, write, edit ACL, ...

Opening a file

inserts file as segment in address space

Problems with ACLs

The ACL may grow larger than the file

Store ACL in a file?

Create named groups of users?

ACL not accurate record of access

presence of directories adds complexity

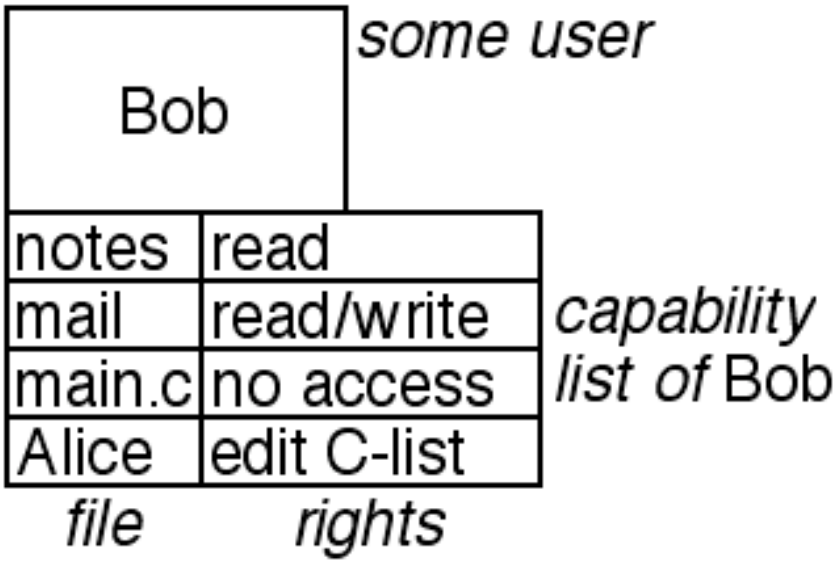
ACL's are undisciplined

Users can do anything

Freedom = responsibility

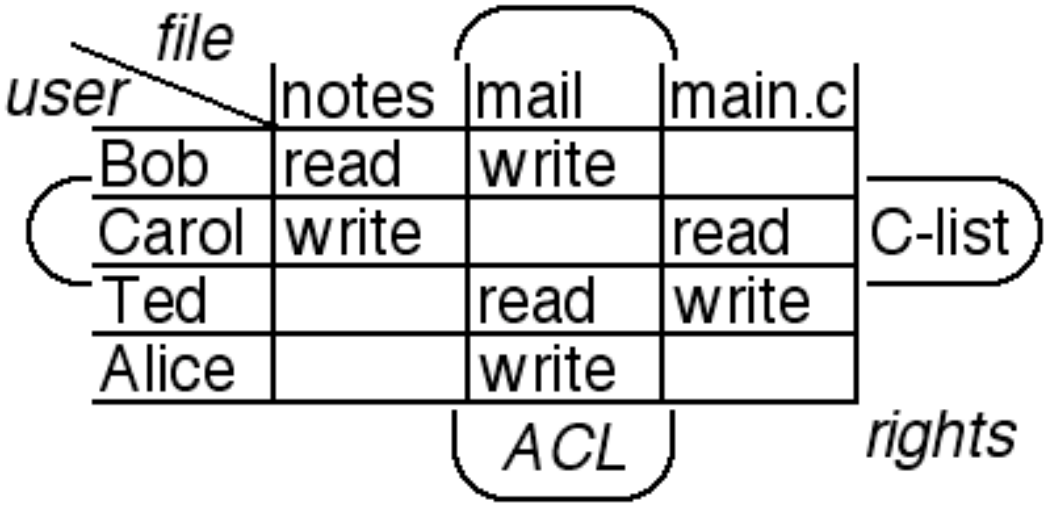
Capability Lists

Dennis and Van Horn, CACM, March 1966



Are C-lists duals of ACLs?

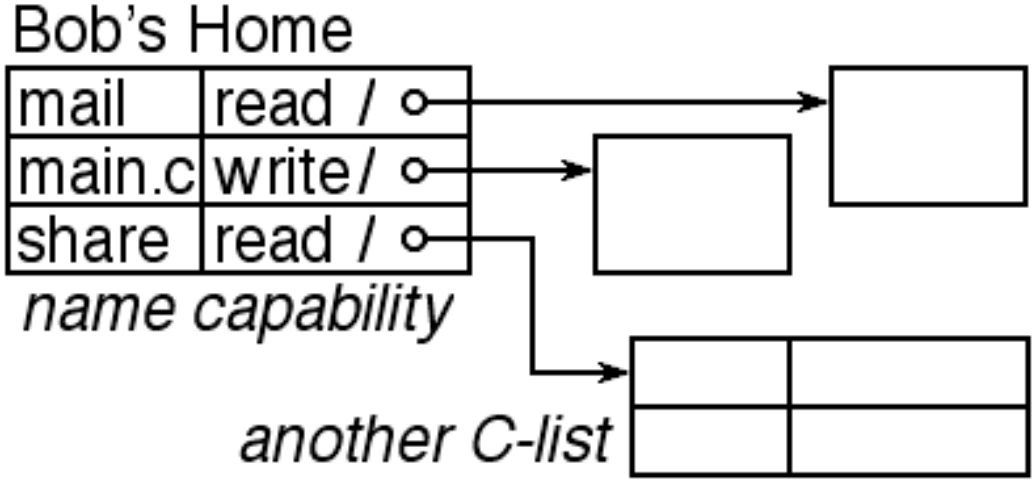
The Access Matrix:



Capability-based addressing

C-List = Directory !!!

Fabry, CACM, 1974



The Unix file system

Files have 3-entry ACL

Owner, Group, Other

Directory structure is hierarchic

except for uplinks, symbolic links

Ownership does not confer access

Groups

solve many problems but

Group creation is superuser-only

Group manipulation is clumsy