Mar 7, 2005 -- Lecture 20

THE UNIVERSITY OF IOWA

22C:169 Computer Security

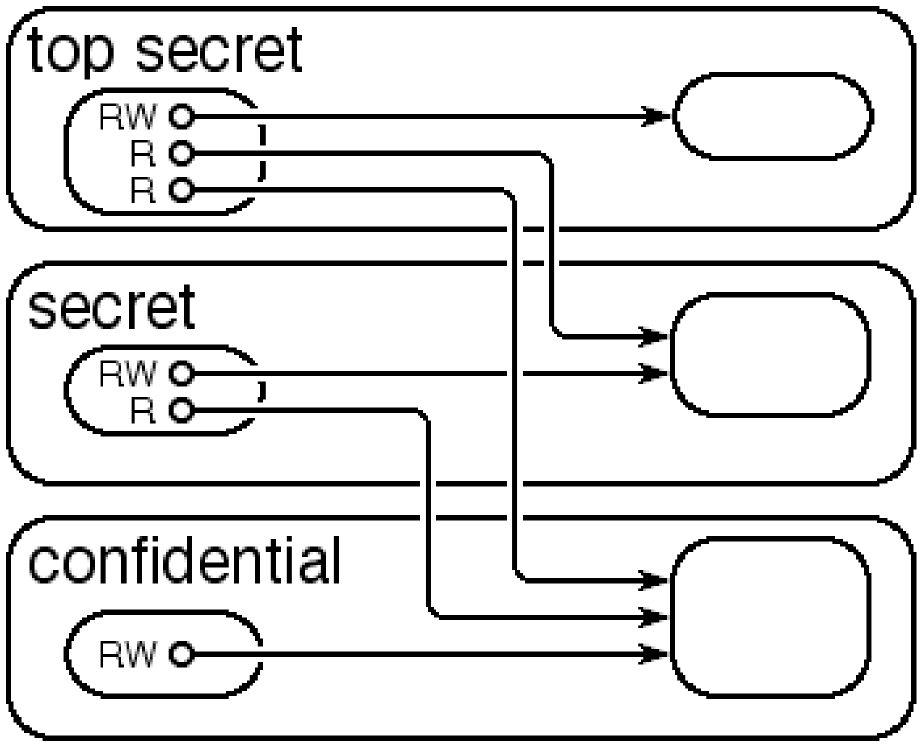
Douglas W. Jones

Department of Computer Science

Design Problems

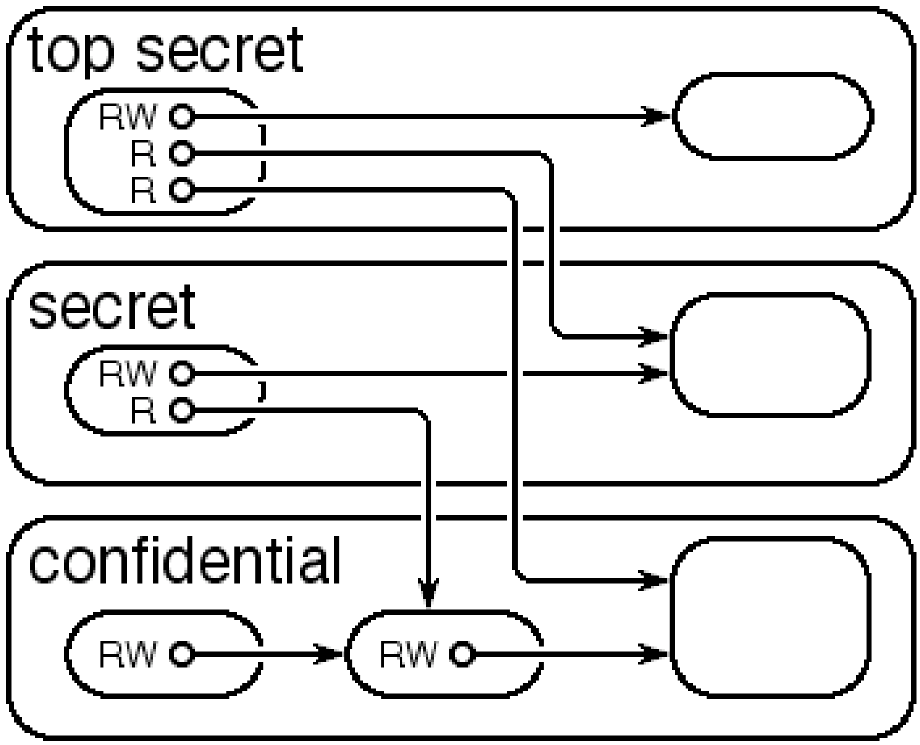
Hierarchic rules in a Capabilty System

Easy if no capabilities for C-lists



Hierarchic rules in a Capabilty System

Hard if capabilities for C-lists



Solutions

Label each object/user with classification

Add support for this to kernel

Orthogonal to capabilities

Consequences

RW cap for object may grant no access!

Extra mechanism enlarges kernel

Extra check raises enforcement cost

Solutions II

Introduce path rights

AK Jones, 1973, Carnegie-Mellon Hydra system

Idea

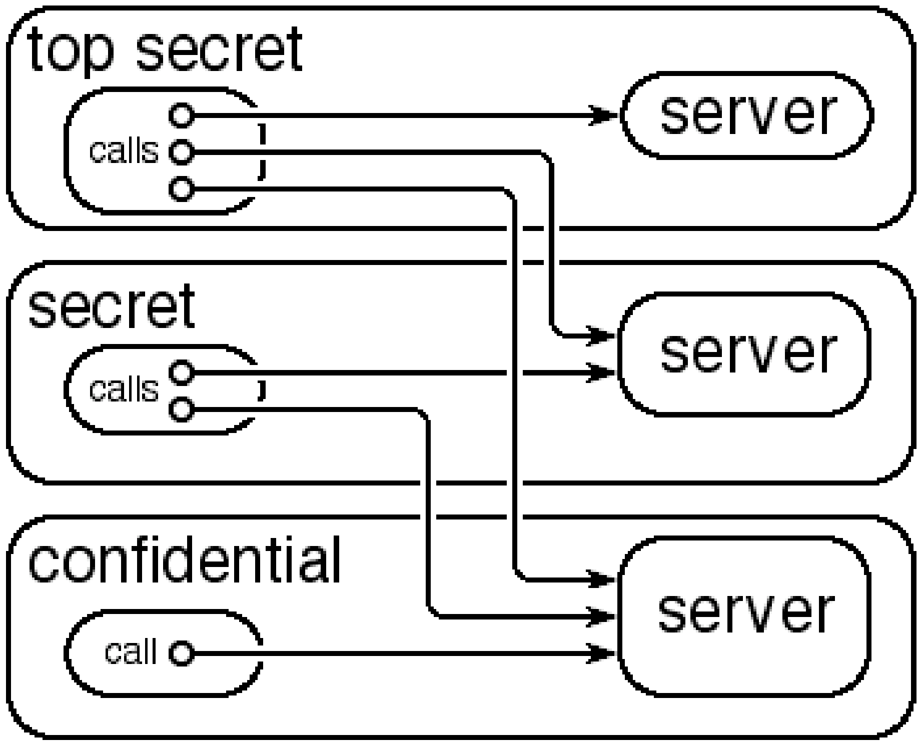
Following path through C-lists confers rights on result that are formed by and of rights in each capability used

Problems

RO C-list for address space that has RW capabilities for variables

Need path rights distinct from access rights to C-list to solve this

Client Server Systems



Problems

Server at same level as client

- *No problem*
    - If we ignore taxonomic issues

Server below level of client

- *Client may not send request!*

Conclusions:

- *client-server illegal under Bell Lapuda?*
- *or servers must all be in trusted core?*
    - Either is distressing

Conclusion

Bell-Lapuda model is wrong!

- *We already knew this*
- *Useful systems always violate it!*
  - use of secrets violates hierarchy

Useful security models must

- *Control, not prevent, use of secrets*

Goal: minimize the trusted core

Ideally,

Device drivers outside the core

Servers outside the core

We want these written as user code

In general

We cannot do this

We can, however,

Produce a minimal kernel

Simplify the audits of non-kernel servers

What do we want in trusted servers?

Non-disclosure

Server should not disclose info
Except to legitimate users

Non-examination

Server should not examine data
Except as needed for its function

No covert services or channels

Client-server protocols should carry minimum information

Dangerous Functionality:

Log files, for example, stream of:

`<timestamp, client, service>`

This file can help detect covert abuse

This file can serve as a covert channel!

reading log files is dangerous!

Access to time of day

Needed for recording log entries

Enables creation of timing channels

should be restricted to log server

(other servers pose similar dangers)

Audit of trusted servers

Simplified if domain for each server

Holds only minimum necessary objects

Complicated by use of

Large standard environments

Difficult when

indirection, overloading, etc.
are used to hide identities of servers