Authorisation

Introduction to IT Security

Otago Polytechnic
Dunedin, New Zealand

From last time

AAA

- Authentication: who you are
- Authorisation: what you can do
- Accounting: tracking what you have done

Authorisation: Strategic concerns

A lot of the security literature seems to focus on access to information, i.e., what you can see. This is the wrong way to think about authorisation. Instead, think about controlling access to *actions*. (Think verbs, not nouns.)
Example: My access to the student database

- Action: create a new student record - not authorised
- Action: read your student record - authorsised
- Action: update your student record - not authorsised
- Action: delete your student record - not authorsised

Authorisation: Strategic concerns

Least privilege approach

- The default answer to "Can I do this?" should be "No".
- Selectively choose to say "Yes" only when the requestor
  - is trusted
  - needs access

Authorisation: Policy

The theory is that *management*, not IT, should set authorisation policies. However, in practice

- Managers won't set policy because they doesn't realise that they need to, and
- will get it wrong if they try.

So as IT people we should be prepared to guide management through the policy making process. This doesn't mean that IT should take over. Management should decide, with help from IT.

Authorisation Types: DAC

Discretionary Access Control

- A user has control of certain resources.
- The user may choose to grant some control to other users or groups.
- The grants are generally tracked with an *Access Control List* (ACL).
- DAC is the access control method used by nearly all operating systems.

```
tclark@biblios:~$ ls -l /tmp
total 16
-rw-r--r-- 1 tclark staff     0 Sep  5 11:06 examplefile
drwx------ 2 tclark tclark 4096 Sep  5 08:07 i3-tclark.KUjqAK
drwxr-xr-x 3 root   root   4096 Sep  5 08:05 passenger.1.0.2488
drwx------ 2 tclark tclark 4096 Sep  5 10:17 pulse-FdFIn2LtSoNz
drwx------ 2 tclark tclark 4096 Sep  5 08:07 ssh-8C0VROBLuELE
```

Authorisation Types: MAC

Mandatory Access Control

- All resources are tagged with security attributes.
- All users are also tagged with secutiry attributes.
- Whenever a user attempts to access a resource, the secutity attributes of the user and resource are checked to see if the access is authorised.
- Attributes and policies are manged by a security administrator. Users can not override or changes security attributes and policies.
- Examples include SELinux, FreeBSD.

Authorisation Types: RBAC

Role-Bases Access Control

- Users are assigned to a number of *roles*, perhaps by job function.
- Access to resources is permitted for users who have the allowed role.
- Example: Practically every web content mangement system.

Authorisation Types: RBAC[1]

Rule-Bases Access Control

- The access policy is built around a number of rules that govern access.
- Examples:
  - Users may be able to access a resource during certain hours of the day.
  - Users may be able to access a resource from specified network locations.

[1] The same intialism twice. Wow, that sucks.

Today's Lab

Design an R(ole)BAC scheme for a new College of EAD Content Management System.

- The system will allow staff to create, edit, pubilsh, unpublish, and remove web pages.
- The system will have four sections: Whole College, BIT, BAM, and Certificate Programs.
- Ordinary staff will be able to create and edit page drafts in their area.
- Team leaders will be able to publish drafts so that they appear on the web site and unpublish theem.
- The Head of College, Leslie, and her designates shall have global rights to do anything.

Your task is to identify a set of roles for this system.

- List the roles.
- Identify which staff may be in which roles. (N.B.: A person may have multiple roles)
- Make a (preliminary) list of the tasks associated with each role.