Foundations of Computer Security

Lecture 27: Storing the ACM

Dr. Bill Young
Department of Computer Sciences
University of Texas at Austin

Access Control Matrix

Recall our earlier claim: Any access control policy can be represented by an *access control matrix* (ACM).

	object$_1$	...	**object**$_k$
subject$_1$	A_i, A_j		$\emptyset$
...			
subject$_n$	A_l		A_i, A_m

The ACM gives an explicit representation of every access permitted by every subject to every object.

Representing Access Information

You *could* build an explicit ACM for any access control system (e.g., BLP, Biba, RBAC, etc). But we usually don't. *Why not?*

Three common alternatives exists:

1. Maintain a set of rules to compute access permissions "on the fly" based on attributes of subjects and objects.
2. Store the permissions with objects. This is called an *access control list* (ACL).
3. Store the permissions with subjects. This is called a *capability-based system*.

Access Control List (ACL)

	Object $_j$
Subject $_1$	RW
Subject $_2$	R
⋮	⋮
Subject $_n$	X

An *access control list* (ACL) stores permissions with the objects of the system.

It contains pairs of the form $< S, P >$, listing the set of permissions P that subject S currently holds to the object.

Any request by subject S for access A to object O, means checking whether $A \in P$ for the pair $< S, P >$ on O's access control list.

Unix/Linux, Mac OS, and Windows all store permissions by ACL.

```
drwxr-s--x 2 byoung prof   4096 2011-06-30 16:49 graphics
-rw-r----- 1 byoung prof 135269 2011-07-05 16:36 lecture20.pdf
-rw-r----- 1 byoung prof 126135 2011-07-05 16:36 lecture20.ps
-rw-r----- 1 byoung prof  42375 2011-07-06 11:28 lecture20.tex
```

Capabilities

Some systems store permissions with subjects rather than objects. These are called *capabilities*.

	Obj $_1$	Obj $_2$		Obj $_k$
Subject $_n$	R	RW	•••	W

Each subject S maintains a collection of pairs $< O, A >$, meaning that S has current permission to perform access A to object O. To obtain access, the subject must present an appropriate capability. Thus a capability is a type of "ticket."

Many capability based systems also permit passing capabilities from one subject to another, under controlled circumstances.

Protecting Capabilities

Possession of a capability is *de facto* evidence of permission. Therefore, no access check is required. But to maintain security, it is necessary to ensure that capabilities can't be *forged* or *altered*.

Historically, various approaches have been used to protect the integrity of capabilities:

- Extend each memory location with an additional bit indicated whether or not the location contains a capability; only the OS can manipulate capabilities.
- Store capabilities in specially protected memory.

Lessons

- Any access control system can be represented by an access control matrix.
- Storing the matrix explicitly is expensive and usually unnecessary.
- Access information is often stored: implicitly as a series of rules, with each object as an access control list, or with each subject as a collection of capabilities.

Next lecture: Information Theory