CS 194-1 (CS 161) Access Control

Doug Tygar (doug.tygar@gmail.com)

September 2, 2005

cs161.org

Role of Access Control

- Before closing "back doors" we need to close "front doors"
- Access control: determines access to files & processes in OS
- We will return to these themes throughout the course

Classic models of security

- Computer security has its origin in military models of security
- Different levels of secrecy
  - e.g. classified/secret/top secret
- Compartmentalized security
  - e.g. nuclear, communications, etc.
  - TS/SCI

Corresponding access control

- Classic model → Mandatory Access Control (MAC)
  - (we also use the abbreviation MAC for "message authentication code")
- User controlled security → Discretionary Access Control (DAC)

Subjects & Objects

- Subjects do things
- Objects have things done to them
- Access types are the things that are done

Subjects & Objects

- Subjects do things
  - users, processes …
- Objects have things done to them
  - files, processes …
- Access types are the things that are done
  - read, write, append, list, detect, remove, execute …

Read and write are different

- Access types can be distinguished by whether they pass information
- Generally "write" passes information
- Generally "read" does not pass information

Access Control Matrix

	File 1	File 2	File 3
Alice	read		read/write
Bob		execute	
Charlie			read

Problems with access control matrix

- Sparse matrix – many blank entries
- Hard to manage
- Who can manage different entries?
- What if we need to give "temporary rights"?
- Common entries?

Sparse matrix representations

- Access Control Lists (ACLs)
  - objects lists subjects and access types
  - example: this file can be modified by Alice and read by Charlie
- Capabilities
  - subjects have particular "permissions"
  - example: Bob is allowed to modify files
- Hybrid models also exist

ACLs & Capabilities: equivalent?

- In representative power, yes
  - Both are sparse matrix representations of the Access Matrix
- In philosophy, no
  - Often come with particular features & OS philosophy
  - Capabilities often appeal to researchers
  - But capability systems often work poorly
  - Perennial claim: Capability lists are coming back!

Where is ACL applied?

- Some systems: on the file
- Some systems: on the directory
- Some systems: combination

Who determines identity?

- In (non-distributed) multi-user systems, usually OS
  - login
- In distributed systems
  - Sometimes a central authority
  - (trusted third party, e.g., Kerberos)
    - Single login
  - Sometimes knowledge of a password
  - (e.g., ssh or "guest" file sharing in Windows)
    - Remote login

Who is allowed to modify ACL?

- In some systems, the "owner" of the file/process/directory
- Example: chmod command in UNIX
  - World access: read/write/execute
    - For directories: read = list items;
    - execute = "enter" directory
  - Owner access: read/write/execute

Fine-grained control

- But we need options other than "world access" or "owner-only access"
- General ACLs allow arbitrary access, but hard to manage
- Solution: groups

Groups

- A group is a single id such as
  "Berkeley-undergrads"
  "friends of Alice"
  "administrative access"
- A group administrator maintains group membership list

More on UNIX chmod

- World: read/write/execute
- Group: read/write/execute
- Owner: read/write/execute
- Can change owner using chown command

Temporary access

- This is an area where capabilities systems excel
  - "transferring a capability"
  - Sometimes like giving a reference
- ACL systems need special mechanism
  - UNIX: "setuid"
  - Windows NT/XP: "run as"

Procedure-oriented access control

- We run a program to determine access
- Example: Web server access

Monotonic vs. non-monotonic

- Classic access control was monotonic
- As we acquire more capabilities, or identities, we get more powerful
- "Root", "Super-user", "Administrator"
- But this often causes problems
  - What if "root" password is discovered?

Non-monotonic access control

- In non-monotonic access control, as we gain identities or capabilities, we may lose access
- Example: Windows file sharing (administrators have crippled access)

Distributed access control

- Distributed access control is an active research area
- Example: who can access an encrypted satellite broadcast?
  - Users join and leave all the time
  - Millions or tens of millions of users
- "Distributed key distribution"

Compatibility of access control

- ACLs predominate, but each system implements them in their own way
- Systems must "translate" access control
  - SAMBA supports Windows and Unix-like systems
- Continual source of serious errors

Autonomous access control

- Each system manages its own access control
- Requires remote login
- Problem: people often access hundreds or thousands of systems, and necessarily reuse login info (passwords)
- Common password problem
- We will revisit these issues in the course

Access control is central to security

- We return to access control repeatedly in the course
- Old area of security, but not well understood
- Often poorly implemented
- And we haven't even begun to look at "backdoors"!