Windows Access Control

#access_control/windows #access_control/acl

Windows Access Control

- access control on objects
  - can define **arbitrary classes**
  - **active directory** - defines new classes
- classical protection system
  - full-blown *access control lists* (even negative ones)
  - discretionary protection state operations
- not so usable, few people have experience

Access Control Lists (ACL)

- *columns* of the access matrix represent **access control lists** for the objects
- **access control list** - list of the principals that are *authorized* to have access to some object
- ACLs for access matrix below
  - $O_1 : S_1$
  - $O_2 : S_1, S_2, S_3$
  - $O_3 : S_3$

	O_1	O_2	O_3
S_1	Y	Y	N
S_2	N	Y	N
S_3	N	Y	Y

Security in Windows

- phases of security development
- early windows systems were based on DOS style security
  - *assumption* of computers being single-user devices
  - with the internet, became less and less true

- security model today came from improved merging of DOS style security and VMS-like
  - UNIX originally developed for systems with VMS

Major Access Control Parts

- **access tokens** - contains information about logged on user
- **security descriptors** - contains security information about an object
- user *authenticates* with account name and password
  - based on the login the system creates the access token

Access Tokens

- similar to user and group ids in unix
- **security identifier** (SID)
  - SID of user account
  - SID for group associated with user, login, and owner
- **user privileges** - pre-defined set of rights to system resources and tasks
- subsequent processes *inherit* access tokens
- different processes have different rights

Security Descriptors

- provide security information about objects in the os
- **discretionary access control list (DACL)** - specifies access allowed for objects
  - similar to unix mode bits
- **system access control list (SACL)** - ability to access objects
  - administrators only
- **access control entry (ACE)** - element within ACL that contains
  - *security identifier* for object owner
  - *access mask* with access rights for object
  - *flag* showing ACE type
  - *inheritance* bits
- **ACE authorization**
  - ACEs for particular request are *totally ordered*
  - start from the top and check each ACE, authorize for *SIDs* in token on *set of rights*
  - if ACE matches SID (e.g. user, group, login)
    - **deny** - ACE denies access for some specific right

    - **need full coverage** - ACE grants access for some rights
- reach bottom and not all SIDs granted, **deny request**