Operating System Security

CS460

Cyber Security Spring 2010

Outline

- Unix/Linux Access Control
  - Users and groups
  - File system controls
- Windows NT/XP/Vista/7 Security Executive
  - Access tokens
  - Security descriptors
  - ACLs
  - Integrity Controls (Vista)

Unix Reading Material

- Man pages
  - Groups, newgroup
  - Chmod, chown, chgrp
- Unix and Security: The Influences of History

Basic Unix Security Model

- User authenticated on logon
  - User ID associated with process
  - Default Group ID associated with process
  - Default Process listed in passwd file
- Groups defined in /etc/groups
  - Set of users listed with each group definition
  - User can be member of multiple groups

Shadow Files

- /etc/passwords and /etc/group must be readable by everyone
- Both files contain crypt’ed passwords
  - Access enable offline attacks
- Add shadow versions of each file
  - Password obscured in passwords and group
  - Stored in more restricted shadow versions of these files

Unix Access Control

- Three permission octets associated with each file and directory
  - Owner, group, and other
  - Read, write, execute
- For each file/directory
  - Can specify RWX permissions for one owner, one group, and one other

Unix Access Check

- First test effective user ID against owner
  - If match, then use owner rights
- Then test all groups user is a member of against group
  - If match, then use group rights
- Otherwise, use other rights
- Can view as rwx, or a value from 0-7
  - E.g. rx = 5 and rw = 6

Constraining Control of New Objects

- Umask can be set to constrain allowed access on new objects created by user
- Expressed as a 3 octet mask
  - E.g. 0022
- Inverse of umask anded by requested access for new object
  - E.g. open requests 0666 (read and write for all)
  - 0666 & ~0022 = 0666 & 755 = 644

Other Bits

- Set UID and Set GUID bits
  - When set, the process created by executing file takes on user ID or group ID associated with file
- Sticky bit
  - On directories, prevents anyone but owner of file removing file in directory

Unix Security Problems

- Created as a subset of more complete Multics model
  - Expedient at the time
  - Limits modern expressibility
- Security evolved over 30 years
  - Inconsistencies
- Early evolution occurred in open university environments
  - Encourages bad habits

Windows Reading Material

- Windows NT Security in Theory and Practic
  - Old, but still a readable introduction
- Windows Access Control
  - Newer version of above
- Inside Windows NT Chapter 3 or Microsoft Windows Internals Chapter 8
- Windows Vista Integrity Mechanism
- Vista Security Features

NT Security Model

- Ultimately NT security controls access and auditing
- Implements the standard subject/object security model
  - Designed into NT. Implemented a security reference monitor
- Controls applied to core OS objects like processes and sockets in addition to the more tradition file system elements (NTFS)
  - Everything that can be named is an object
  - All objects can have same security controls applied

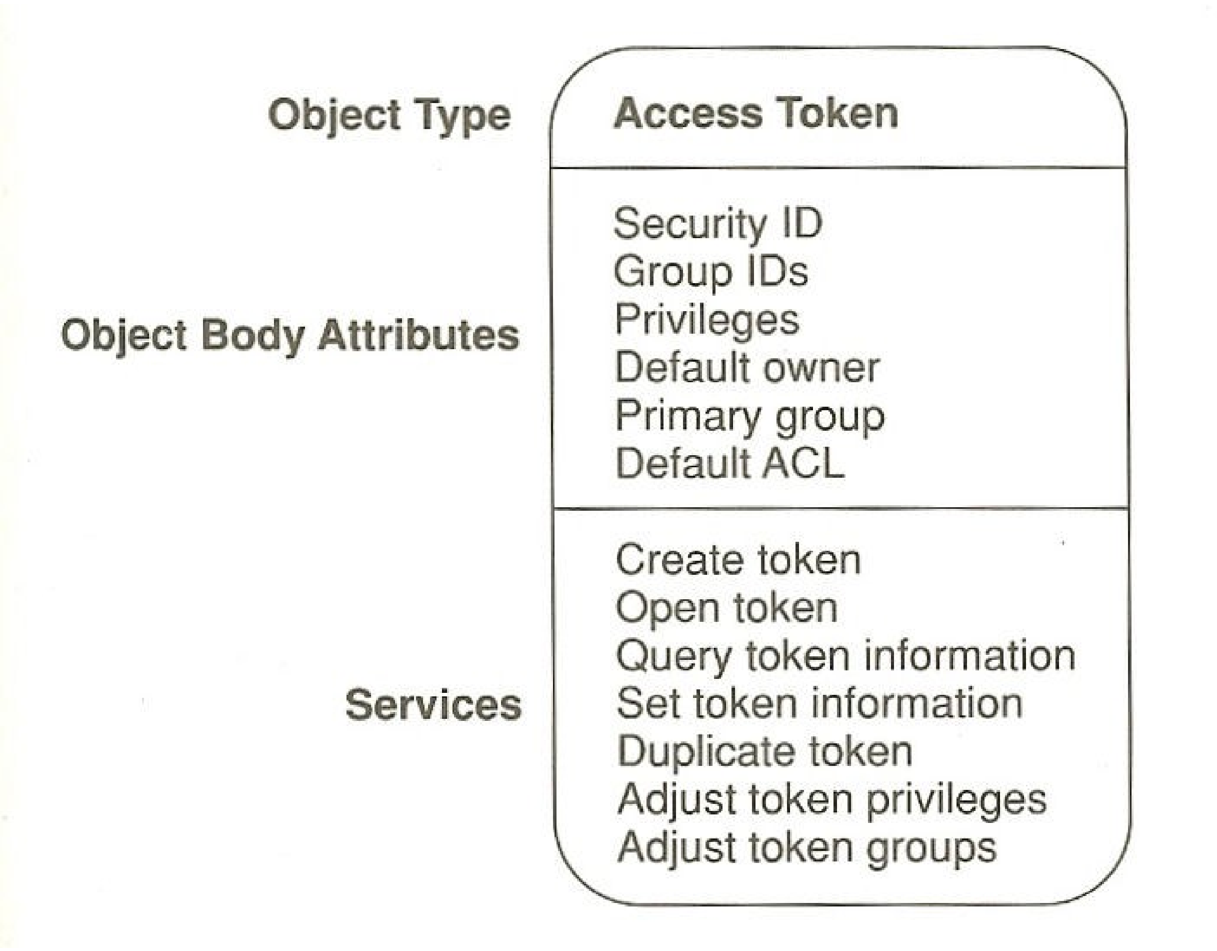
NT Security Elements

- Subject – Process or thread running on behalf of the system or an authenticated user
- Security ID (SID) – A globally unique ID that refers to the subject (user or group)
- Access token – the runtime credentials of the subject
- Privilege – ability held by the subject to perform "system" operations. Usually breaks the standard security model
  - Associated with the access token
  - Generally disabled by default.
  - Can be enabled and disabled to run at least privilege
  - Example powerful privileges
    - **SeAssignPrimaryTokenPrivilege** – Replace process token
    - **SeBackupPrivilege** – Ignore file system restrictions to backup and restore
    - **SeIncreaseQuotaPrivilege** - Add to the memory quota for a process
    - **SeTcbPrivilege** – Run as part of the OS

Windows User/Group Definitions

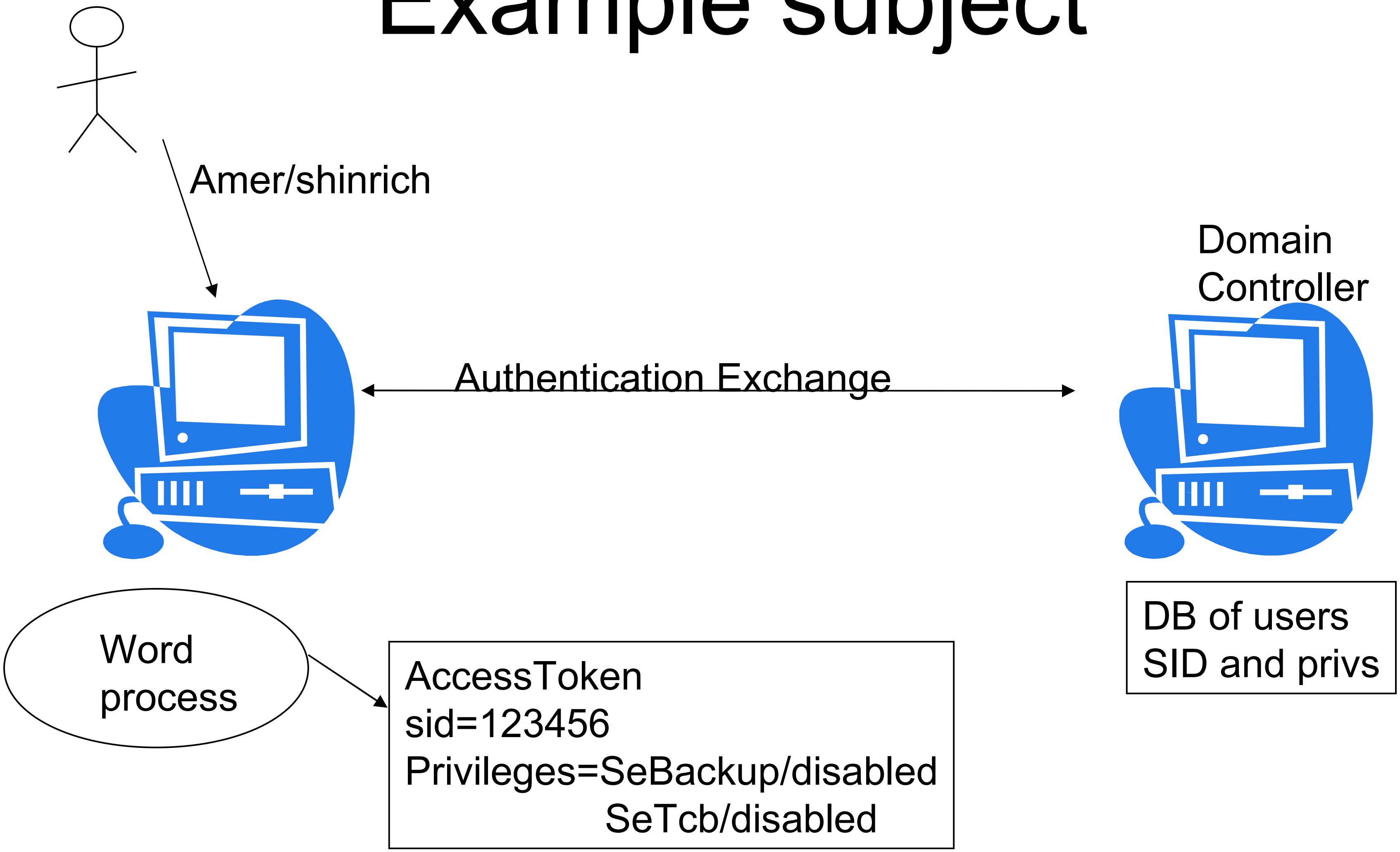
- Control Panel/Computer Management
  - Contains the User/Group definition
- Control Panel/Local Security Settings
  - Under user rights
  - Lets the user associate users and groups with privileges

Access Token



Object Type	Access Token
Object Body Attributes	Security ID Group IDs Privileges Default owner Primary group Default ACL
Services	Create token Open token Query token information Set token information Duplicate token Adjust token privileges Adjust token groups

Example subject



Amer/shinrich

Domain Controller

Authentication Exchange

Word process

AccessToken
sid=123456
Privileges=SeBackup/disabled
SeTcb/disabled

DB of users
SID and privs

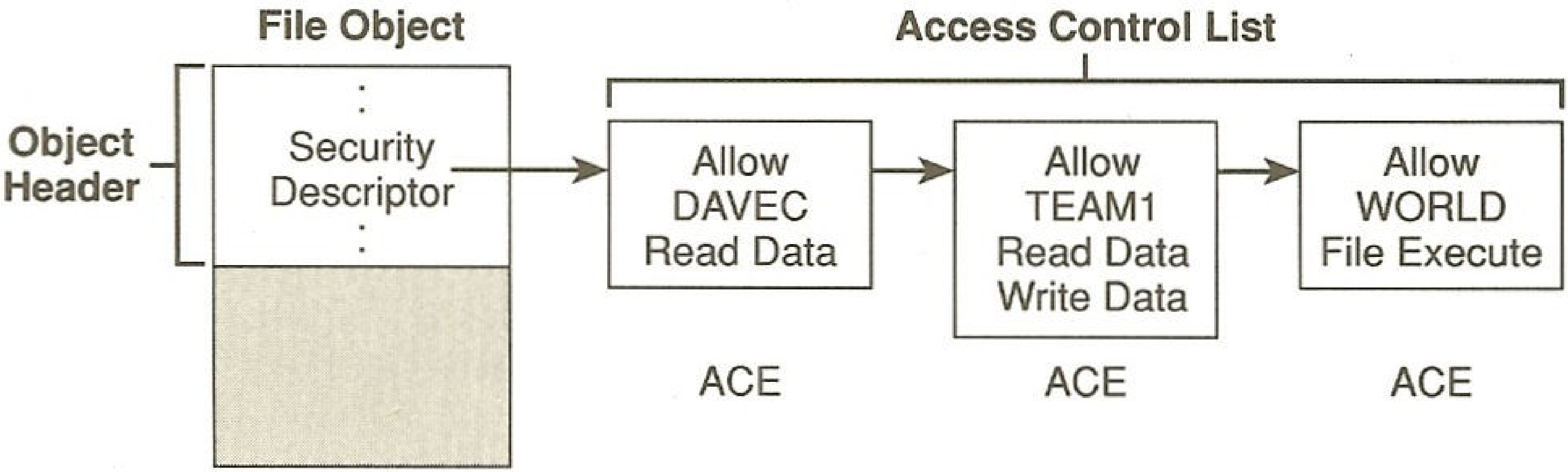
More security elements

- Object – Individually secured entity such as a file, pipe, or even a process
- Rights – actions associated between object and subject.
  - Read, write, execute, audit
- Access control list (ACL)
  - Associated with an object
  - Ordered list
  - Each access control entry (ACE) contains a subject and a right
  - Evaluated by the security subsystem to determine access to protected objects.
  - Discretionary ACLs control access
  - System ACLs control audit (and integrity control)

Still more security elements

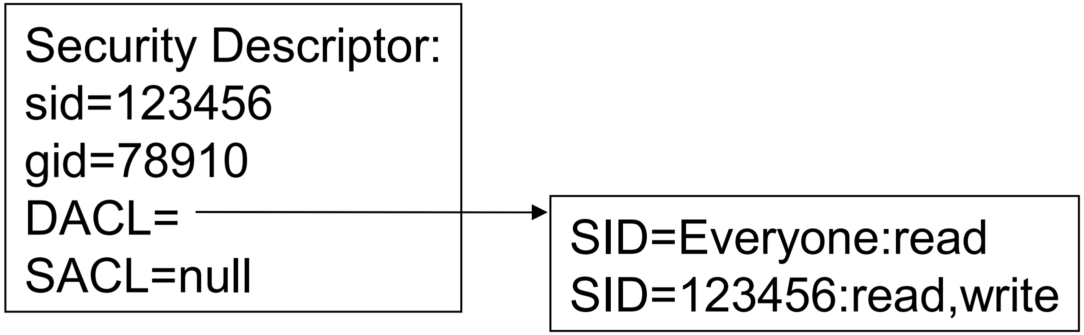
- Security Descriptor – represents an object in the system. Contains the following information:
  - Object’s owner
  - Object’s group
  - Object’s DACL
  - Object’s SACL
- **AccessCheck** evaluates an ACL, subject, object triple
  - Called by many system calls
  - Can be called from user code too

Security Descriptor



Example ACL

\mydocs\hw1.doc

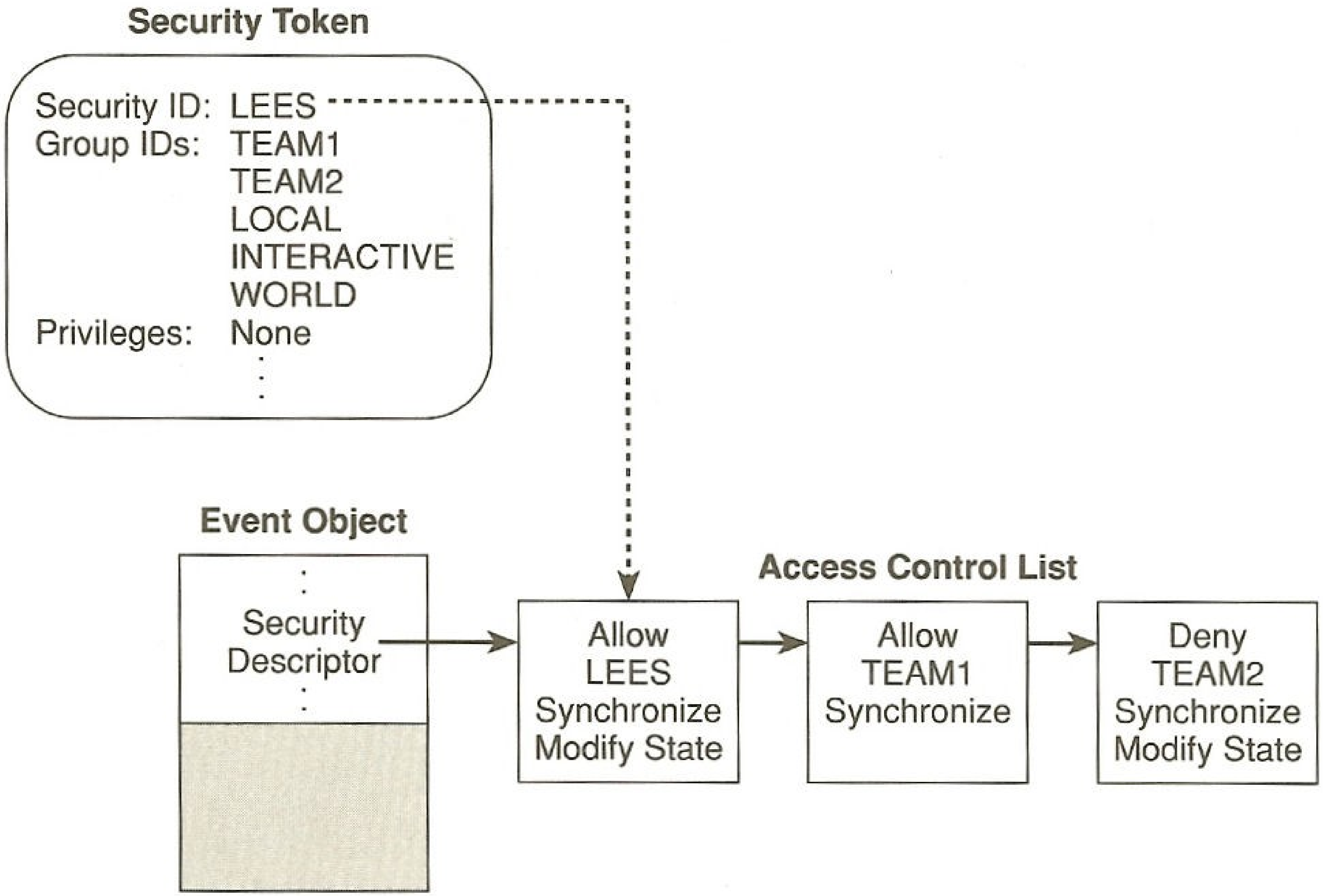


Security Descriptor:
sid=123456
gid=78910
DACL=
SACL=null

SID=Everyone:read
SID=123456:read,write

SID=22222:deny
SID=Everyone:read
SID=123456:read,write

Example Evaluation



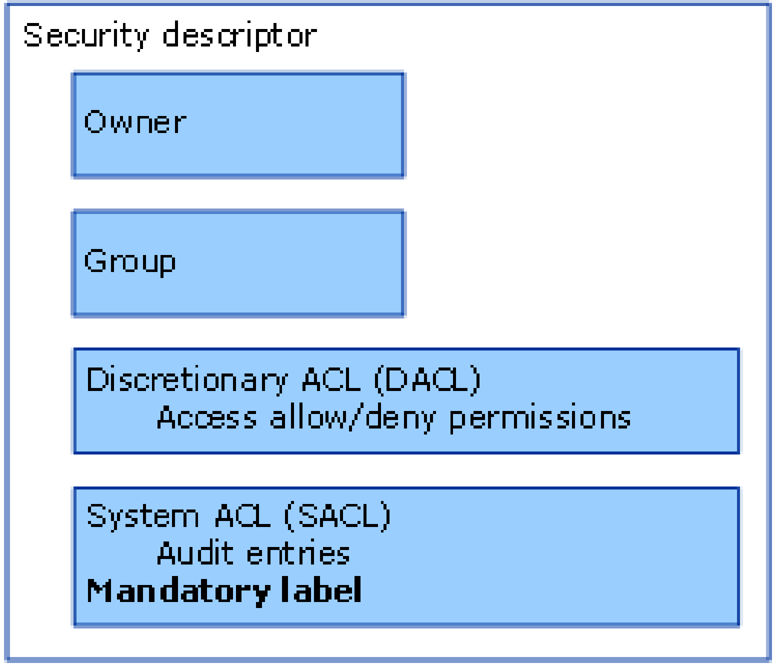
Working with ACLs

- Accessed via FileExplorer. Right-click file/directory an select sharing and security.
- Can programmatically create and traverse ACL’s
  - See MSDN for details

SACL controls auditing

- In addition to DACL that controls access, each object has a SACL to control auditing
  - Process access token is compared to SACL to determine whether to log
  - Also enabled by local policy
- SACL now also includes integrity label

Vista Security Descriptor Plus Integrity Label



Security descriptor

Owner

Group

Discretionary ACL (DACL)
Access allow/deny permissions

System ACL (SACL)
Audit entries
Mandatory label

Mandatory Integrity Controls

- SID representing Integrity Label
  - In Access Token
  - In SACL
- Policy controls execution
  - Mandatory Access Token Policies
    - No Write Up – default- Cannot write higher integrity data
    - New Process Min – default - Controls the label assigned to child processes
  - Mandatory Label Policies
    - No Write Up – default
    - No Read Up
    - No Execute Up

Assigning Token Integrity Label

- Assigned by Group:
  - Local System -> System
  - Administrators -> High
  - Authenticated Users -> Medium
- Some programs designed to run at low integrity
  - Internet Explorer in protected mode -> Low
- Some privileges require integrity
  - e.g., backup, impersonate, relabel

Windows Security Problems

- Kernel level security model is reasonable
  - More consistent and complete than Unix
- So why do Windows installations have so many security problems?
  - Unix evolved from a multi-user environment
  - Windows came from a single user, stand alone environment
  - Security APIs clunky. The easy to program option (NULL DACL) is not the most secure.

Vista Security Additions

- The core security mechanisms are mostly unchanged
  - Addition of mandatory integrity control
  - Dual access tokens
- Important changes in user and service mode
  - Make it easier to run at low privilege
  - User Account Control
- Additional features
  - Host intrusion detection, Firewall improvements, Network quarantine

User Account Control

- Enable non-privileged users to perform many operations that require privilege today
    - Add printer, update WEP keys
- Prompt user to activate privileged account if privilege is needed
- Registry and file virtualization
    - Sandboxes unprivileged users

Windows Service Hardening

- In XP, most services are run as high privilege LOCAL SYSTEM
  - Can run as other user
  - Awkward to install because must create unprivileged user and prompt user to create password etc.
- This create a SID for each service
  - Like an unprivileged user that cannot login

Data Protection

- Uses secure co-processor, Trusted Platform Module, that is included with many of today's laptops
- Use to implement Secure Startup
  - Detects changes to system on reboot
  - Protects from making changes to system made by mounting system from other OS
  - Doesn't seem to have made it into Vista release

Network Access Protection

- Network quarantine
  - Places restrictions on the characteristics of a computer that can connect to the network
  - For example can connect to the network only if the patches are up to date
  - Server version only

Summary

- Standard operating systems security elements
  - Unix shows security has been available for many decades
  - Windows shows security underpinnings exist in widely used OS perceived to be insecure
  - Vista security changes make it easier to use existing security mechanisms
- Security is continuing to evolve