IA 313: Operating Systems Security

Lecture 5: Windows Security

Windows Security

- Windows is the world's most popular O/S
  - advantage is that security enhancements can protect millions of nontechnical users
  - challenge is that vulnerabilities in Windows can also affect millions of users

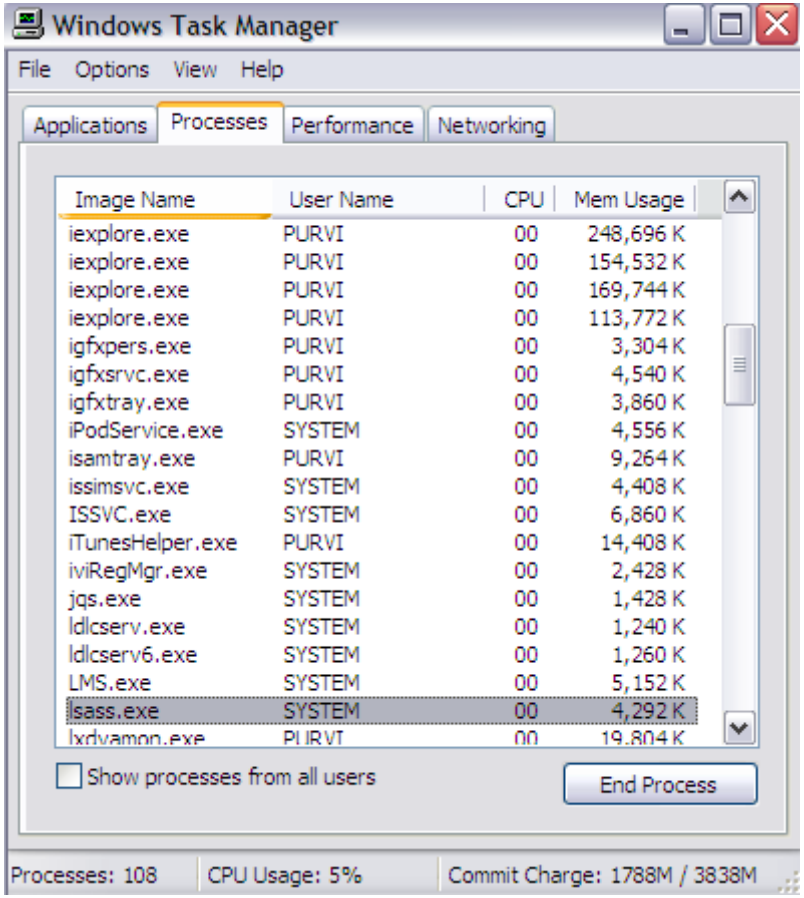
Windows Security Architecture

- Security Reference Monitor
- Local Security Authority
- Security Account Manager
- Active Directory
- Local vs. Domain Accounts
- Access Control Lists
- Integrity Control
- User Account Controls

Security Reference Monitor (SRM)

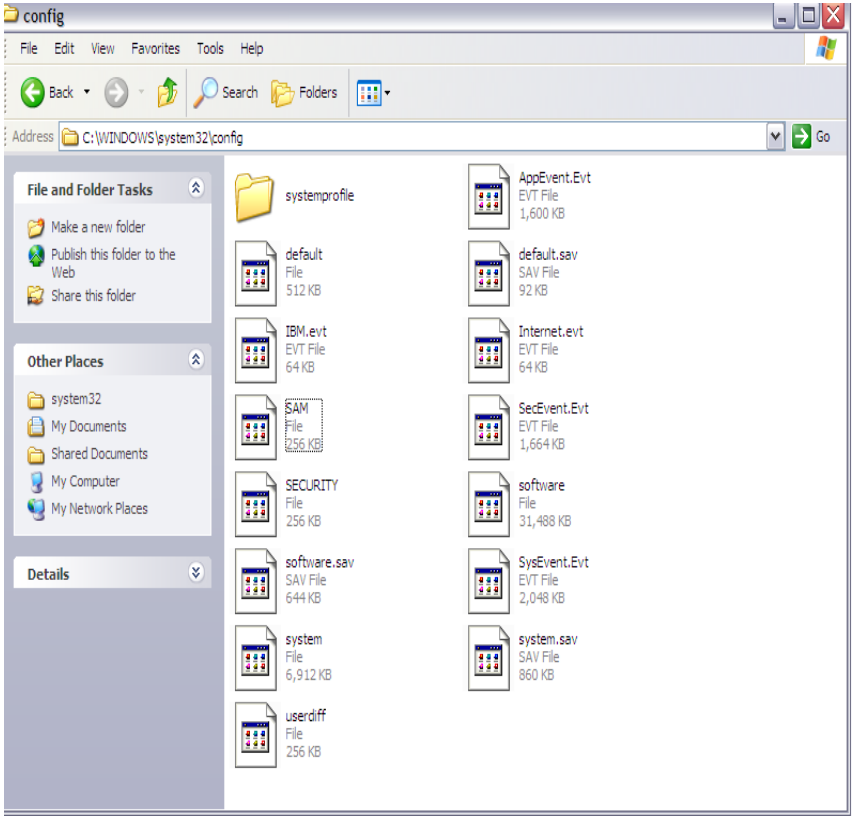
- Kernel Mode Component that
  - Performs Access Checks
  - Generates Audit Log Entries
  - Manipulates User rights (Privileges)

Local Security Authority (LSA)



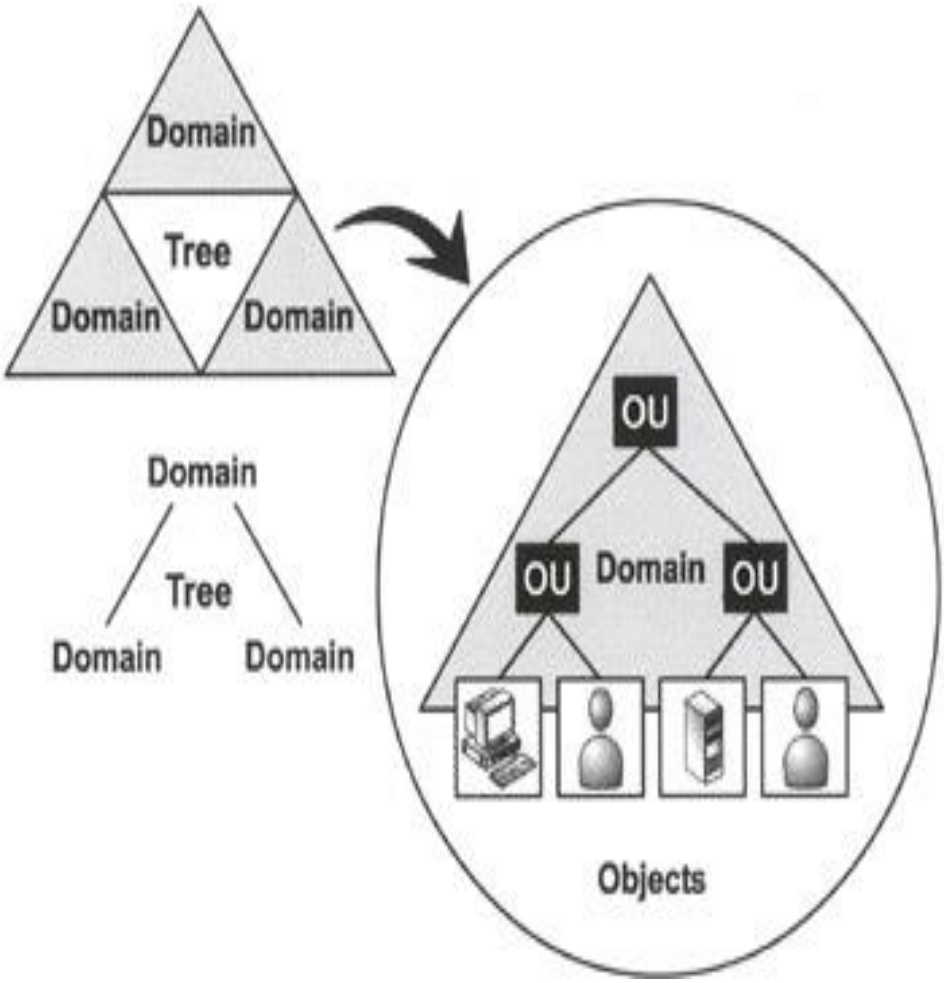
- Responsible for enforcing local security policy
  - Lsass.exe
  - User mode
- Issues security tokens to accounts
- Key component of the logon process

Security Account Manager (SAM)



- A database that stores user accounts and local users and groups security information
- Local logins perform lookup against SAM DB
- Passwords are stored using MD4
- SamSrv.exe

Active Directory



- Active Directory (AD)
  - Microsoft's LDAP directory
  - all Windows clients can use AD to perform security operations including account logon
  - authenticate using AD when the user logs on using a domain rather than local account
  - user's credential information is sent securely across the network to be verified by AD
- Directory Service
  - Server-based authentication
  - Centrally managed

WinLogon & NetLogon

- WinLogon (local) and NetLogon (net) handle login requests
- WinLogon – keyboard requests
- NetLogon – network requests



Local vs. Domain Accounts

- Local Accounts for computers not hooked up to a network
- Networked computers can be:
  - Workgroup joined
  - Domain joined

Workgroup Joined

- A collection of computers connected together
- Only local accounts in SAM can be used
- No infrastructure to support AD

Domain Joined

- Share access to networked printers, file servers, etc.
- Centrally Managed
  - More secure
  - Scalable

Windows Login Example

- Administrator creates a user account (full name, username, password, group, privileges)
- Windows creates an SID in the form of
  - S-1-5-21-AAA-BBB-CCC-RRR
- In windows, username can be in two formats
  - SAM format: support by all versions of Windows (legacy format)
    - Form: DOMAIN/username
  - User Principle Name (UPN) and looks more like RFC822 email address
    - Example: username@domain.company.com

Windows Login Example …

- User logs in with keyboard
- Information is sent to the AD (domain controller)
- If successful token is generated and sent to user
- Token contains
  - User's SID
  - Group membership
  - Privileges

Windows Privileges

- Are system wide permissions assigned to user accounts
  - e.g. backup computer, or change system time
- some are deemed “dangerous” such as:
  - act as part of operating system privilege
  - debug programs privilege
  - backup files and directories privilege
- Others are deemed “benign” such as
  - bypass traverse checking privilege

Access Control Lists

- Two forms of access control list (ACL):
- Discretionary ACL (DACL)
  - grants or denies access to protected resources such as files, shared memory, named pipes etc
- System ACL (SACL)
  - used for auditing and in Windows Vista to enforce mandatory integrity policy

Access Control Lists …

- Objects needing protection are assigned a DACL (and possible SACL) that includes
  - SID of the object owner
  - list of access control entries (ACEs)
- Each ACE includes a SID & access mask
  - Access mask could include ability to:
    - read, write, create, delete, modify, etc
  - Access masks are object-type specific
    - e.g. service abilities are create, enumerate

Mandatory Access Control

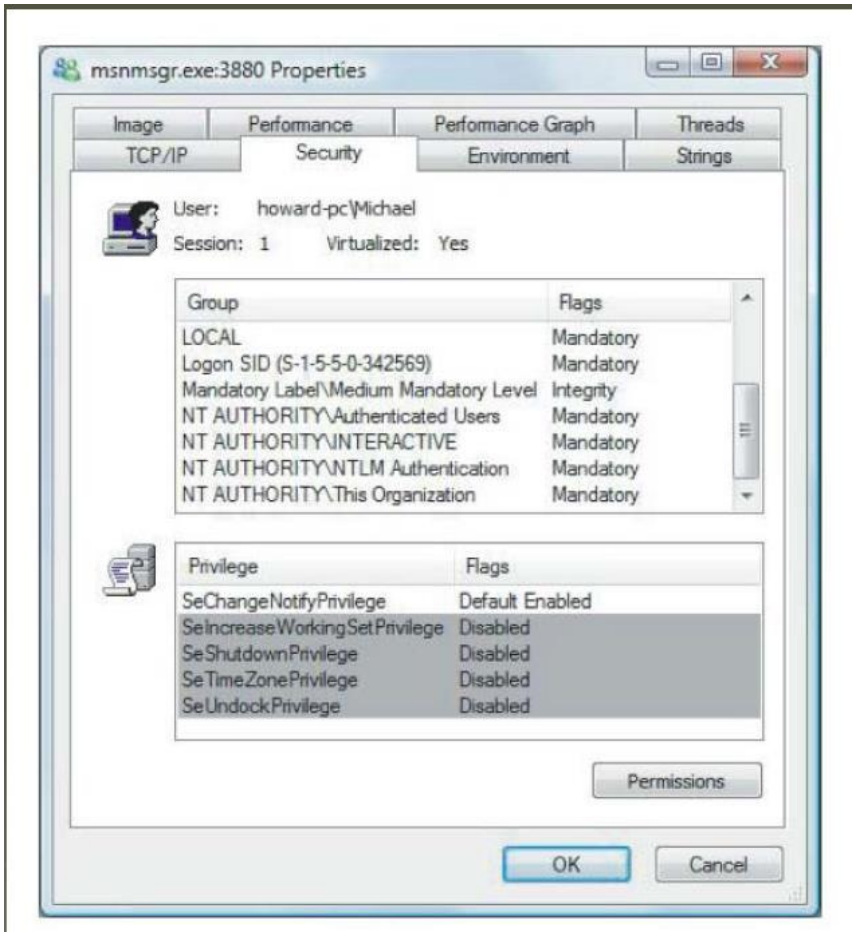
- Have Integrity Control in Windows Vista that limits operations changing an object's state
- Objects and principals are labeled (using SID) as:
  - Low integrity (S-1-16-4096)
  - Medium integrity (S-1-16-8192)
  - High integrity (S-1-16-12288)
  - System integrity (S-1-16-16384)
- When write operation occurs first check subject's integrity level dominates object's integrity level
- Much of O/S marked medium or higher integrity

Integrity Control

- New to Vista: a low-level change to Windows that isolates different objects on a trust-based scale
- Controlled by a new OS component called Windows Integrity Control (WIC)
- Integrity levels trounce permissions
  - Example: malware no longer runs in the privilege level of the logged-on user, as it does in XP
  - It runs in the integrity level of the object that spawned it
- Makes process isolation and other Vista security measures possible

MAC: Vista Integrity Control

- Having Integrity Control in Windows Vista
  - Limits operations changing an object's state



User Account Controls

- A new feature in Windows designed to help prevent unauthorized changes to your computer
- UAC is similar to security features in UNIX-like operating systems
- Perhaps the most reviled and misunderstood feature ever added to Windows



User Account Controls …

- How it works: When your consent is required to complete a task, UAC will prompt you with a dialog box
- Tasks that will trigger a UAC prompt include anything that will affect the integrity or security of the underlying system
  - This is a surprisingly long list of tasks
- UAC works slightly differently with standard user and administrator-class accounts

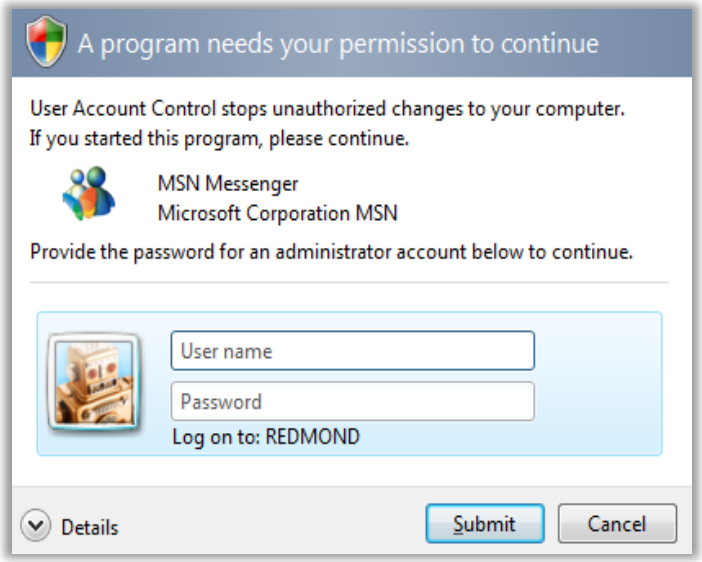
UAC Consent UI: Type 1

- **Prompt:** Windows needs your permission to continue
- **Why you see this:** You attempt to change a potentially dangerous system setting, such as a running a Control Panel



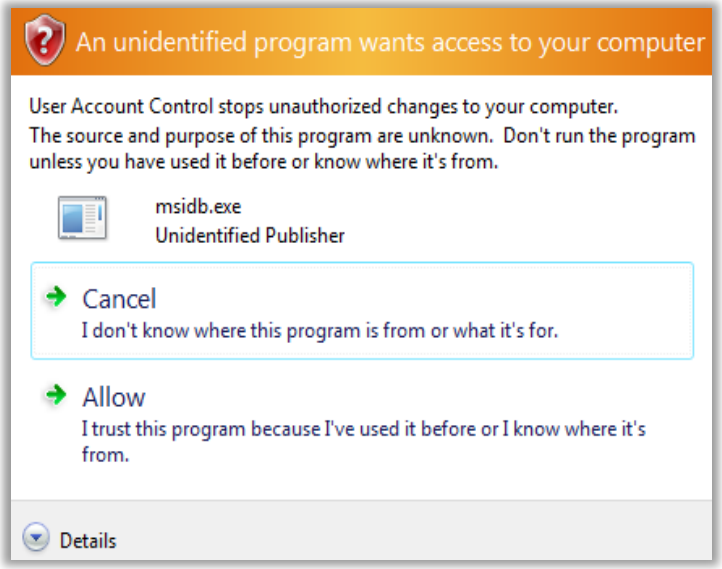
UAC Consent UI: Type 2

- **Prompt:** A program needs your permission to continue
- **Why you see this:** An external application with a valid digital signature is attempting to run with admin privileges



UAC Consent UI: Type 3

- **Prompt:** An unidentified program wants access to your computer
- **Why you see this: i**n external application without a valid digital signature is trying to run



UAC: What's really happening

- Administrator accounts now logon with a mixed token
- Half of this mixed token is a standard user token: this is what is typically used to determine your memberships and privileges
- The other half, the administrator token, is invoked only when required: you can do so manually (run as) or automatically (certain tasks in Vista are tagged as requiring an admin token)

Windows Design Flaws/Poor Design Decisions

- Windows has evolved from a single-user design to a multi-user model few years back
- Windows is monolithic, not modular, by design
- Windows depends too heavily on an RPC model
- Windows focuses on its familiar graphical desktop interface

Evolved from Single-User Design to a multi-user model few years back

- Windows has long been hampered by its origin as a single-user system
  - Windows was originally designed to allow both users and applications free access to the entire system, which means anyone could tamper with a critical system program or file
- Windows XP was the first version of Windows to reflect a serious effort to isolate users from the system, so that users each have their own private files and limited system privileges
  - This caused many legacy Windows applications to fail
  - Solution: Windows XP includes a compatibility mode - a mode that allows programs to operate as if they were running in the original insecure single-user design
- Windows XP represented progress, but even Windows XP could not be justifiably referred to as a true multi-user system

Monolithic by Design, not Modular

- Monolithic Design: one where most features are integrated into a single unit
- Microsoft successfully makes competing products irrelevant by integrating more and more of the services they provide into its operating system
  - But this approach creates a monster of inextricably interdependent services
- Interdependencies side effects:
  - Every flaw in a piece of that system is exposed through all of the services and applications that depend on that piece of the system
  - Unstable by nature: when you design a system that has too many interdependencies, you introduce numerous risks when you change one piece of the system
- Thus, Monolithic system tends to make security vulnerabilities more critical than they need to be

Depends Heavily on an RPC Model

- RPC stands for Remote Procedure Call
- Simply put, an RPC is what happens when one program sends a message over a network to tell another program to do something
- RPCs are potential security risks because they are designed to let other computers somewhere on a network to tell your computer what to do
  - Unfortunately, Windows users cannot disable RPC because Windows depends upon it, even if your computer is not connected to a network
- The most common way to exploit an RPC-related vulnerability is to attack the service that uses RPC, not RPC itself

Focuses on its Familiar Graphical Desktop Interface

- Microsoft considers its familiar Windows interface as the number one benefit for using Windows Server 2003
- By advocating this type of usage, Microsoft invites administrators to work with Windows Server 2003 at the server itself, logged in with Administrator privileges
  - This makes the Windows administrator most vulnerable to security flaws, because using vulnerable programs such as Internet Explorer expose the server to security risks

Windows Vulnerabilities

- Windows, like all OSs, has security bugs
  - and bugs have been exploited by attackers to compromise customer operating systems
- Microsoft now uses process improvement called the Security Development Lifecycle
- Windows Vista used SDL from start to finish
- IIS v6 (in Windows Server 2003) had only 3 vulnerabilities in 4 years, none critical

The core SDL requirements

- Mandatory security education
- Secure design requirements
- Threat modeling
- Attack surface analysis and reduction
- Secure coding requirements and tools
- Secure testing requirements and tools
- Security push
- Final security review
- Security response

Windows Security Defenses

- Attackers are now criminals rather than young, anarchic miscreants, and are highly motivated by money
- Have categories of security defenses:
  - account defenses
  - network defenses
  - buffer overrun defenses.
  - browser defenses

Account Defenses

- User accounts can have privileged SIDs
- Least privilege dictates that users operate with just enough privilege for tasks
- Windows XP users in local Administrators
  - for application compatibility reasons
  - can use “Secondary Logon” to run apps
  - also restricted tokens reduce per-thread privilege
- Windows Vista reverses default with UAC
  - users prompted to perform a privileged operation
  - unless admin on Server

Low Privilege Service Accounts

- Windows services are long-lived processes started after booting
  - many ran with elevated privileges
  - but many do not need elevated requirements
- Windows XP added Local Service and Network service accounts
  - allow a service local or network access
  - otherwise operate at much lower privilege level
- Windows XP SP2 split RPC service (RPCSS) in two (RPCSS and DCOM Server Process)
  - example of least privilege in action

Stripping Privileges

- Another defense is to strip privileges from an account soon after an application starts
  - e.g. Index server process runs as system to access all disk volumes
  - but then sheds any unneeded privileges as soon as possible
  - using AdjustTokenPrivileges
- Windows Vista can define privileges required by a service
  - using ChangeServiceConfig2

Network Defenses

- Need more than user defenses
- Vulnerable to attack via network service
- Have IPSec and IPv6 with authenticated network packets enabled by default in Windows Vista
  - IPv4 also enabled by default, expect less use
- Have built-in software firewall
  - block inbound connections on specific ports
    - Vista can allow local net access only
  - optionally block outbound connections (Vista)
  - default was off (XP) but now default on (Vista)

Buffer Overrun Defenses

- Many compromises exploit buffer overruns
- Windows Vista has “Stack-Based Buffer Overrun Detection (/GS)” default enabled
  - source code compiled with special /GS option
  - does not affect every function; only those with at least 4-bytes of contiguous stack data and that takes a pointer or buffer as an argument
- Defends against “classic stack smash”

Buffer Overrun Defenses …

- No eXecuteNamed (NX) / Data Execution Prevention (DEP) / eXecution Disable (XD)
  - prevent code executing in data segments
  - as commonly used by buffer overrun exploits
  - applications linked with /NXCOMPAT option
- Stack Randomization
  - randomizes thread stack base addresses
- Heap-based buffer overrun defenses:
  - add and check random value on each heap block
  - heap integrity checking
  - heap randomization

Browser Defenses

- Web browser is a key point of attack
  - via script code, graphics, helper objects
- Microsoft added many defenses to IE7
  - ActiveX opt-in
    - unloads ActiveX controls by default
    - when any then first run prompts user to confirm
  - protected mode
    - IE runs at low integrity level (see earlier)
    - so more difficult for malware to manipulate O/S

Cryptographic Services in Windows

- Low-level crypto for encryption, hashing, signing
- Encrypting File System (EFS)
  - allows files / directories to be encrypted / decrypted transparently for authorized users
  - generates random key, protected by DPAPI
- Data Protection API (DPAPI)
  - manages encryption key maintenance protection
  - keys derived in part from user's password
- BitLocker Drive Encryption
  - encrypts an entire volume with AES
  - key either on USB or TPM chip