CSCD 303
Essential Computer Security

Fall 2017

Lecture 6 -

Windows Security

– Access Control/Authentication

Reading: Chapter 5, CompTIA text

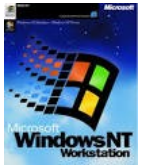
Overview

- Learning Objectives
  - History of Windows Security
  - Learn about Windows Security Components
    - Authentication
    - Access Control
    - Privileges

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
- Review overall security architecture of Windows 2000 and later
  - Look at Security defenses built into Windows

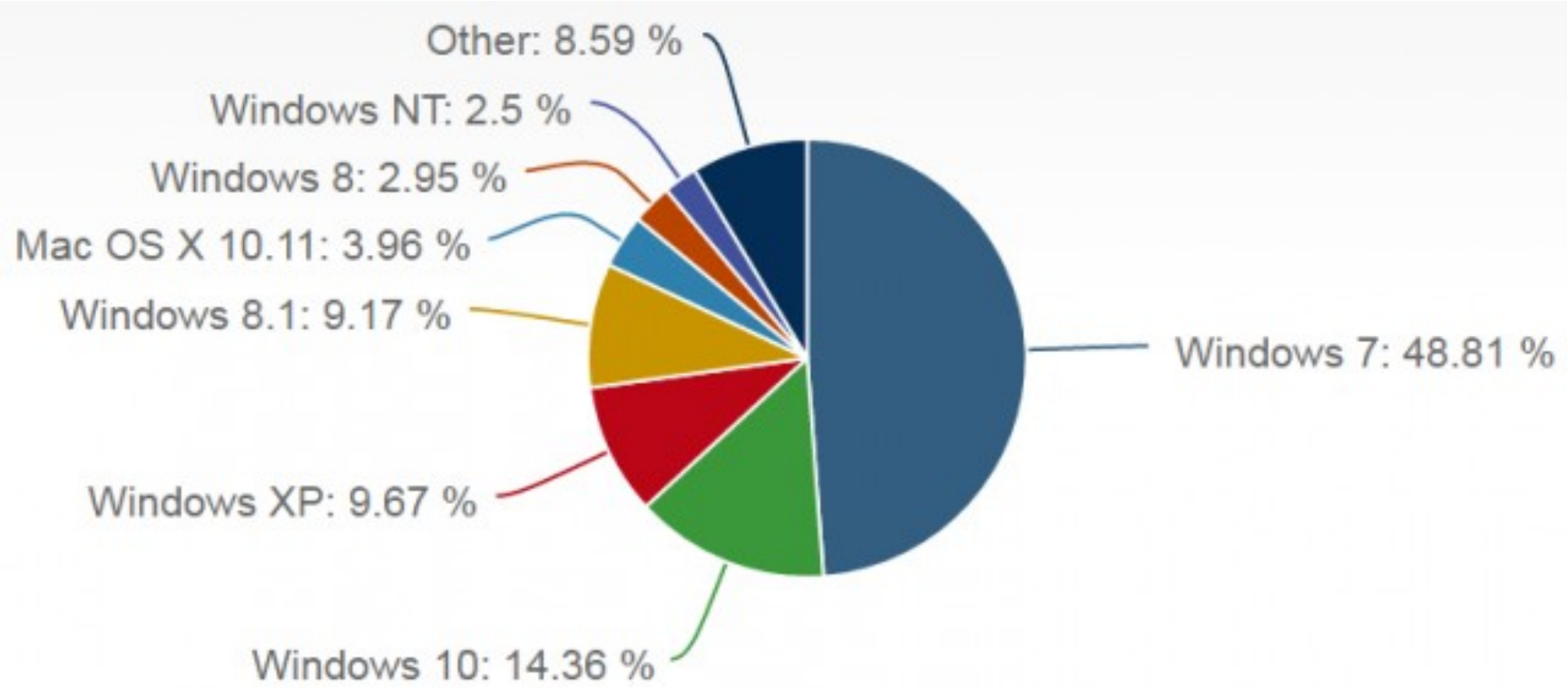
History



- In 1988, Microsoft created “new technology” (NT) portable operating system - supported both OS/2 and POSIX API's. NT supported servers and desktop workstations.
- During development NT was changed to use the Win32 API, reflecting the popularity of the Windows 3.0 Win16 API
- Windows XP was released in 2001 - replace earlier versions of Windows based on MS/DOS, such as Windows98 and Windows ME.
- Windows XP updated in 2005 to provide support AMD64 compatible CPUs, bringing support for 64-bit desktop systems.
- Windows Vista was released in late 2006, but was poorly received due to initial problems with application and device compatibility
- Windows 7 was released in late 2009, greatly improving on Vista.
- Windows 8 was released in October 2012. Not well received.

Windows 10 was first released on July 29, 2015. Windows 10 introduces "universal apps"; Apps can be designed to run across multiple Microsoft product families with nearly identical code

Operating System Market Share



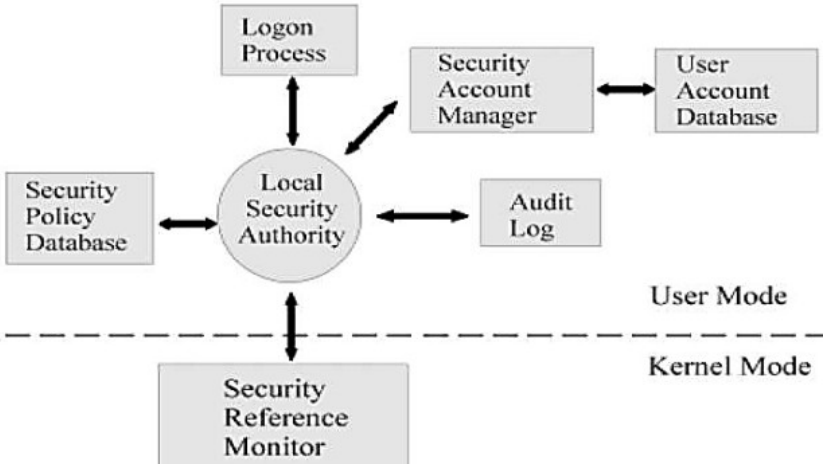
Background: Windows

- **Advantages**
  - User friendly
  - Enhancements can help millions of users
  - Defects found quickly because of widespread use

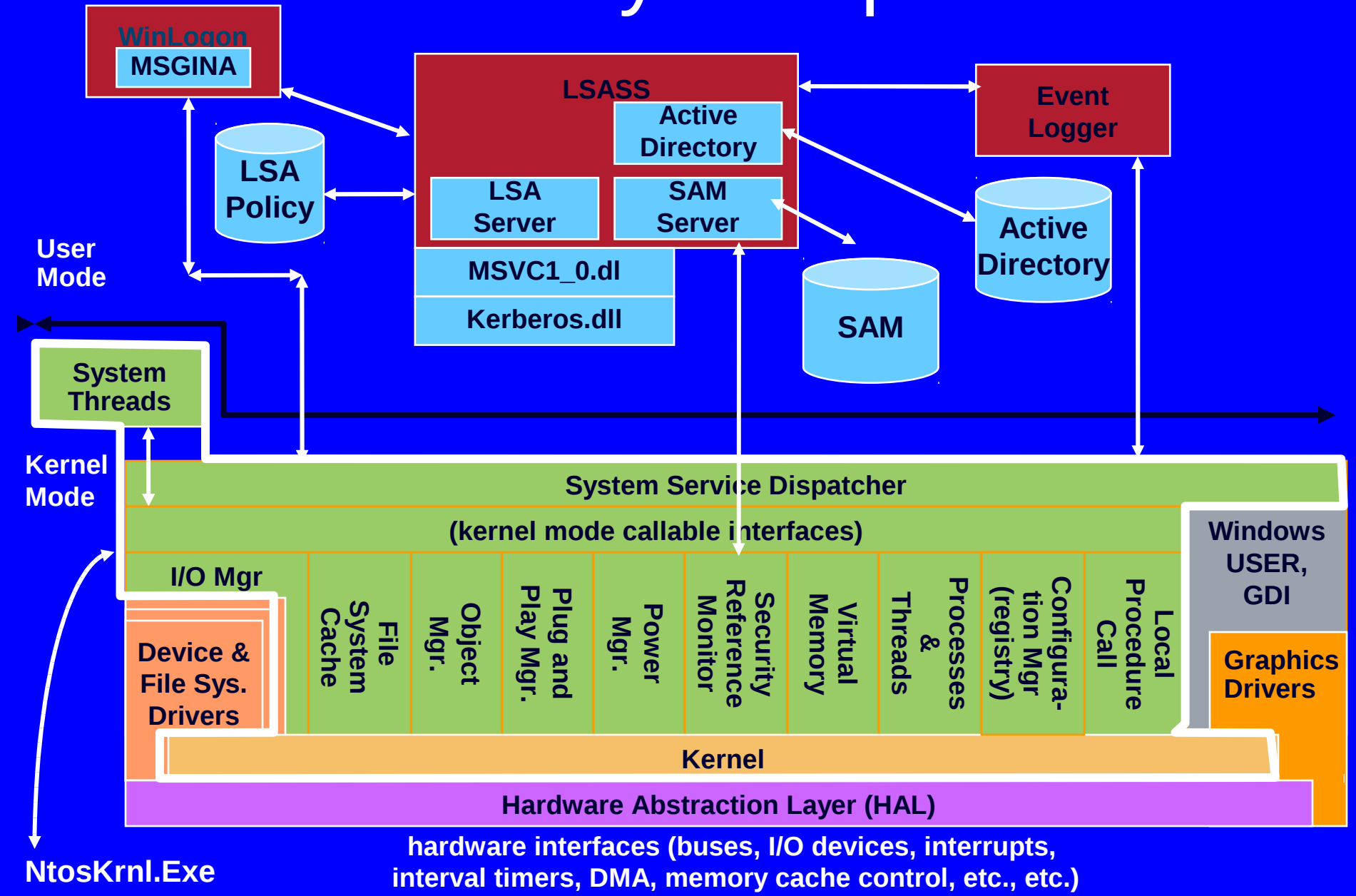
- **Disadvantages**
  - Security defects can leave millions vulnerable
  - Non-technical user-base
  - Industry dominance leaves MS handcuffed - any move to expand capabilities seen as anticompetitive
  - Forced to support legacy code and systems

Windows Security Architecture

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



Security Components



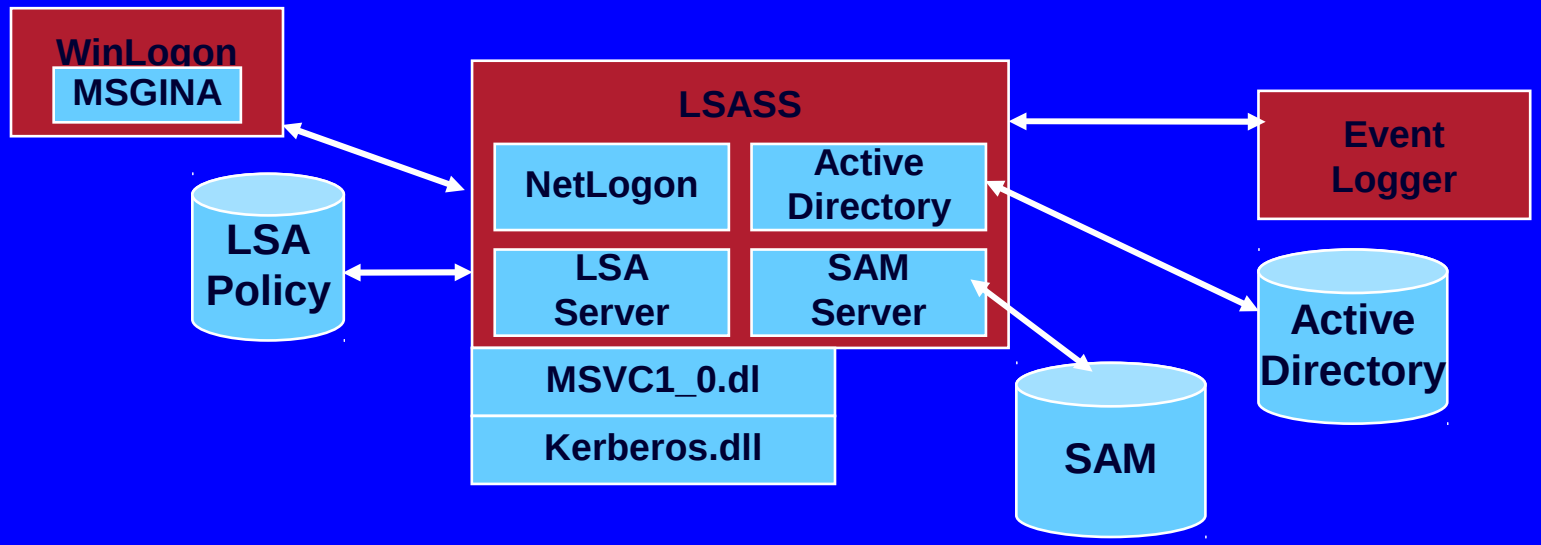
Security Reference Monitor (SRM)

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

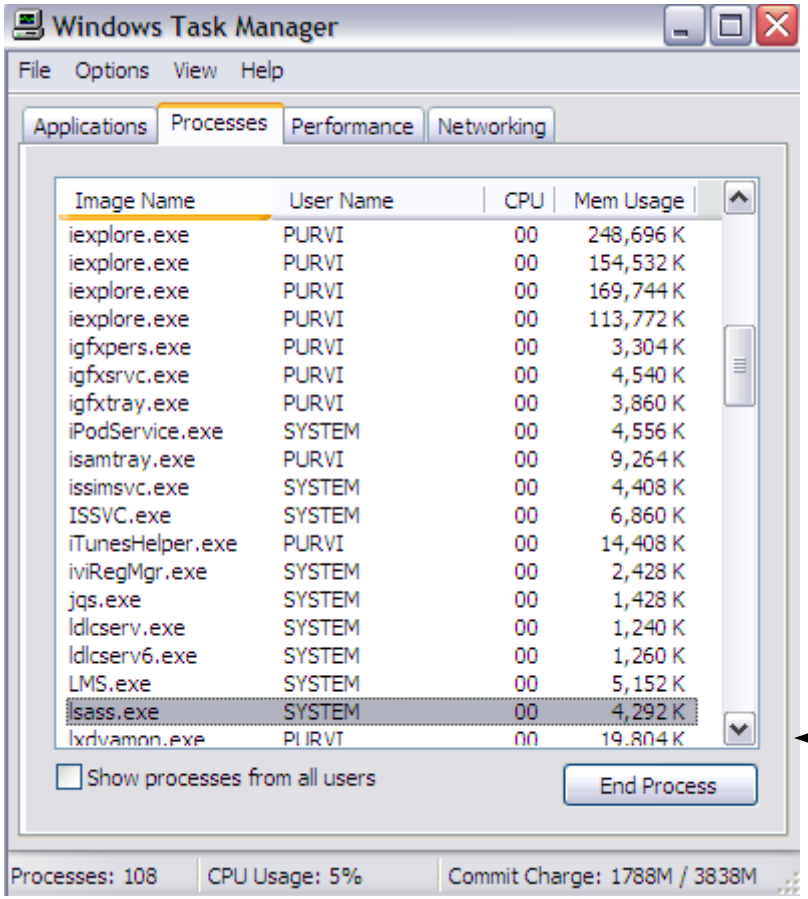
- Group of functions in Ntoskrnl.exe
  - Some functions documented in DDK
    (Windows Driver Devolopment Kit)
  - Exposed to user mode by Windows API calls

Security Components

- Local Security Authority
- User-mode process (\Windows\System32\Lsass.exe) that implements policies (e.g. password, logon), authentication, and sending audit records to the security event log
- LSASS policy database: registry key HKLM\SECURITY



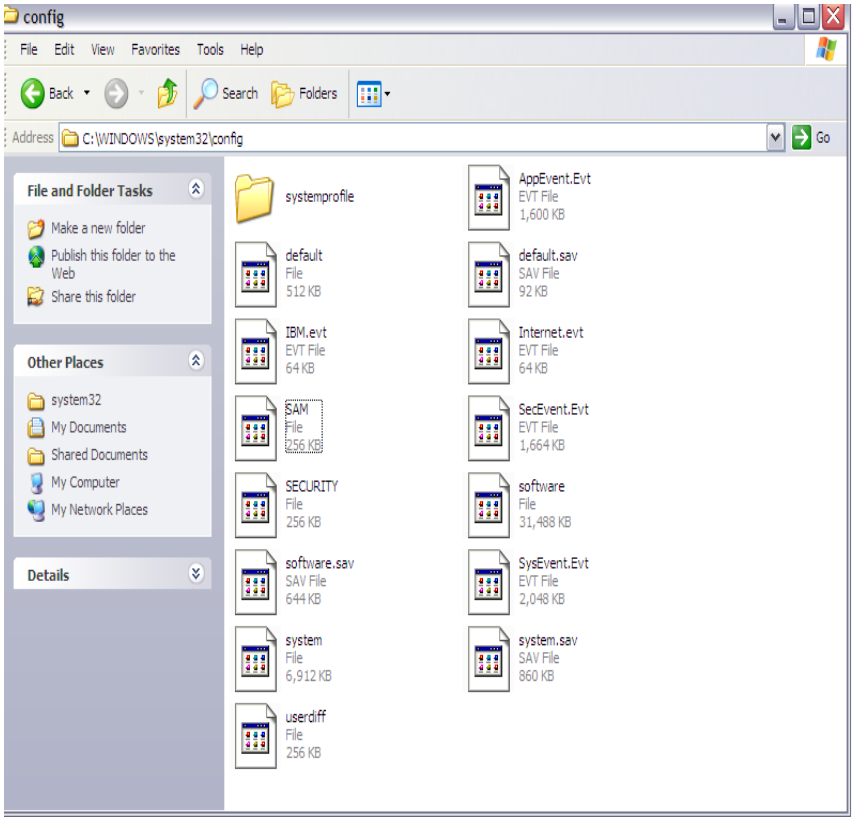
Local Security Authority (LSA)



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

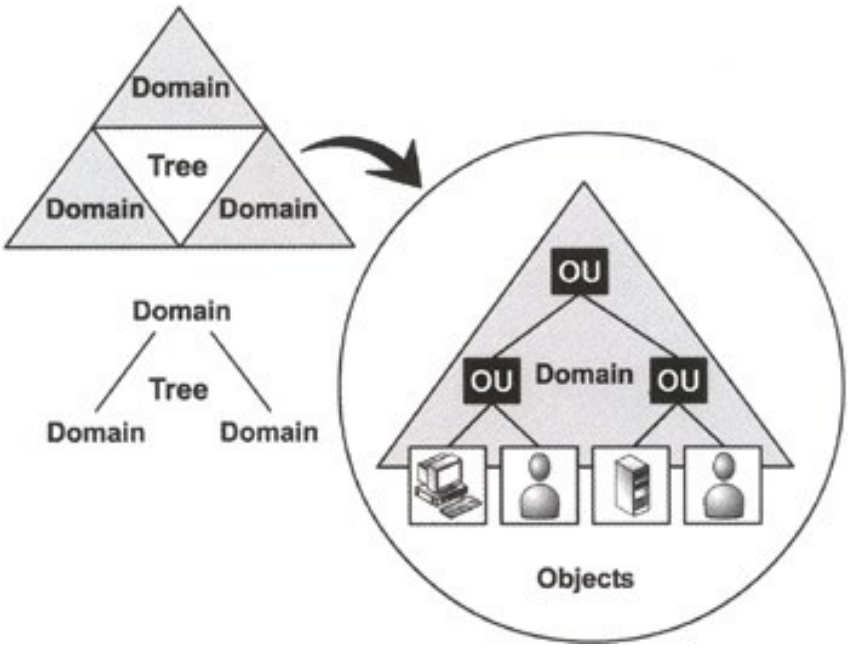
Continuously runs as a process

LSASS Components
Security Account Manager (SAM)



- A database that stores user accounts and local users and groups security information
- SamSrv.exe

LSASS Components
Active Directory



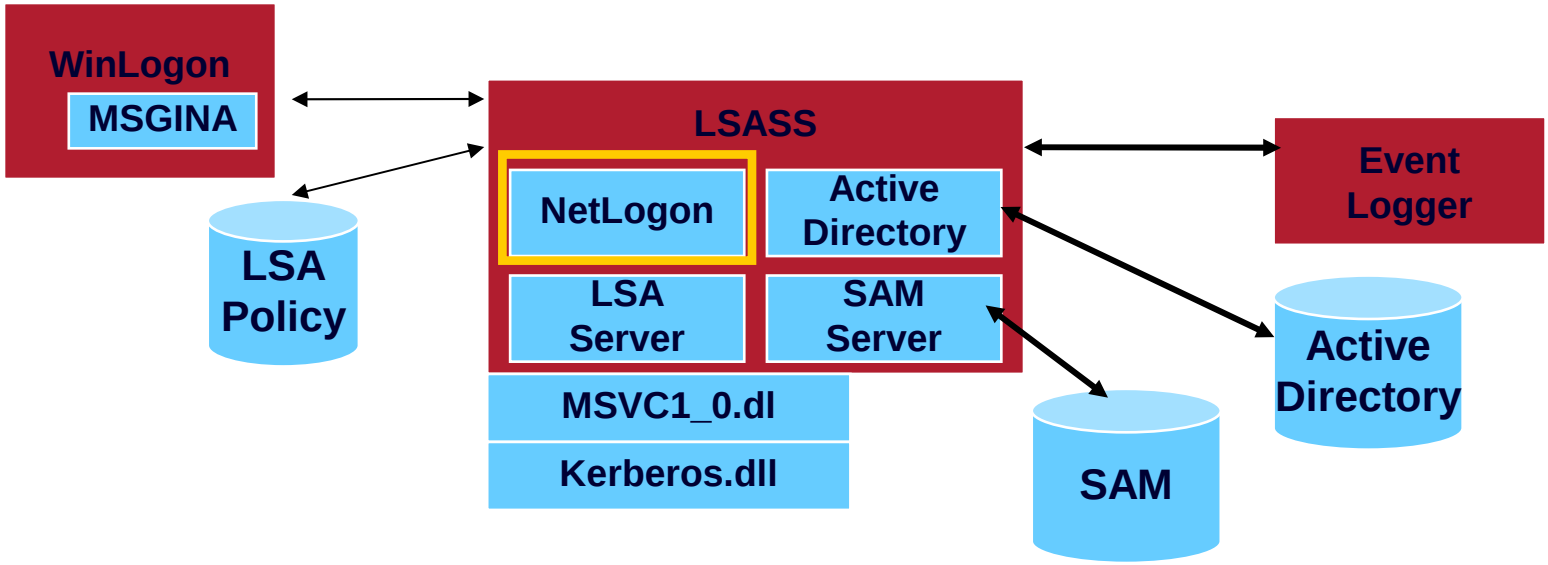
- Directory Service
  - Server-based authentication, non-local
  - Centrally managed by a domain controller

LSASS Components

- Active Directory
  - A directory service contains database that stores information about objects in a domain
  - A *domain* is a collection of computers and their associated security groups that are managed as a single entity
  - Active Directory server, implemented as a service, \Windows\System32\Ntdsa.dll, that runs in the Lsass process
- Authentication packages
  - DLLs that run in context of Lsass process and that implement Windows authentication policy:
    - LanMan: \Windows\System32\Msvc1_0.dll
    - Kerberos: \Windows\System32\Kerberos.dll
    - Negotiate: uses LanMan for local machine or Kerberos, for domain machine

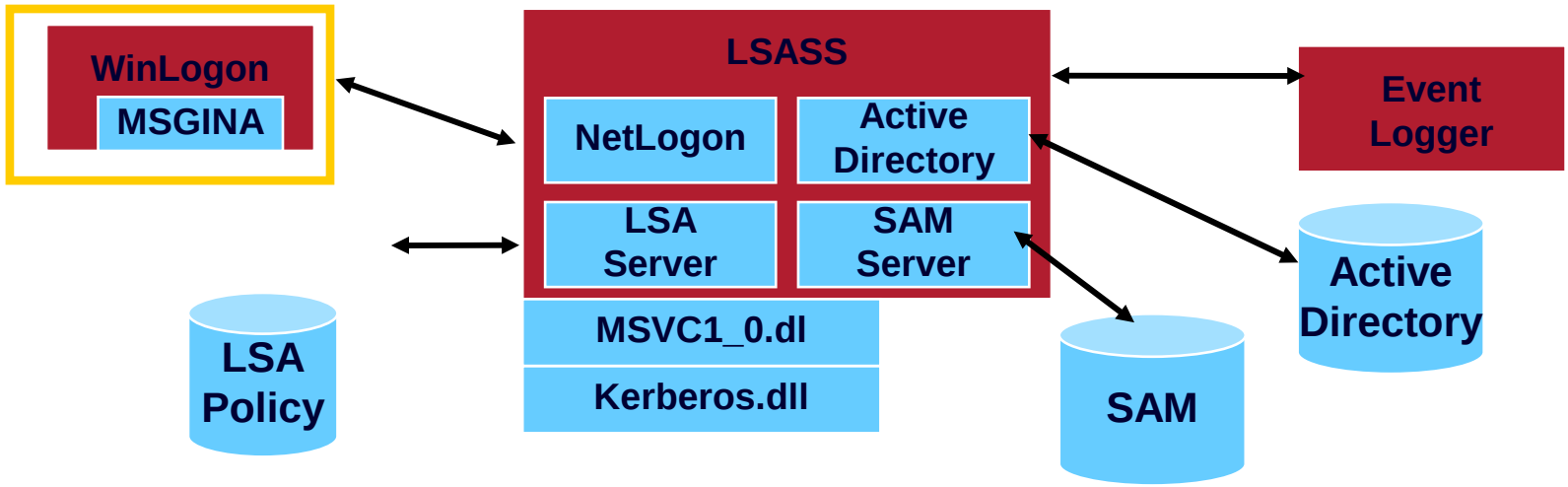
LSASS Components

- **Net Logon Service (Netlogon)**
  - A Windows service (\Windows\System32\Netlogon.dll) that runs inside Lsass and responds to Microsoft LAN Manager 2 Windows NT (pre-Windows 2000) network logon requests
  - Authentication is handled as local logons are, by sending them to Lsass for verification
  - Netlogon also has a locator service built into it for locating domain controllers

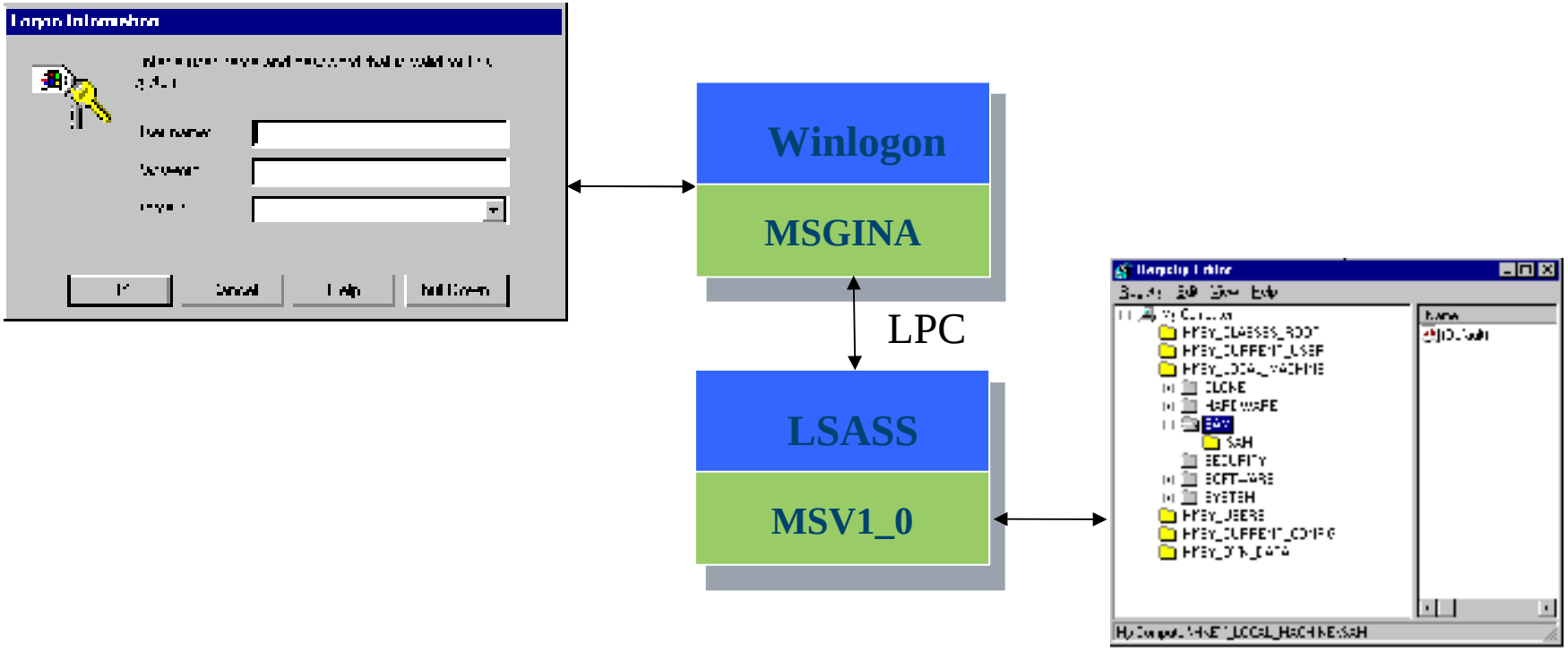


Security Components

- **Logon process (Winlogon)**
  - A user-mode process running \Windows\System32\Winlogon.exe that is responsible for responding to the LSASS and for managing interactive logon sessions
  - Graphical Identification and Authentication (GINA)
  - A user-mode DLL that runs in the Winlogon process and that Winlogon uses to obtain a user's name and password or smart card PIN - Default is \Windows\System32\Msgina.dll

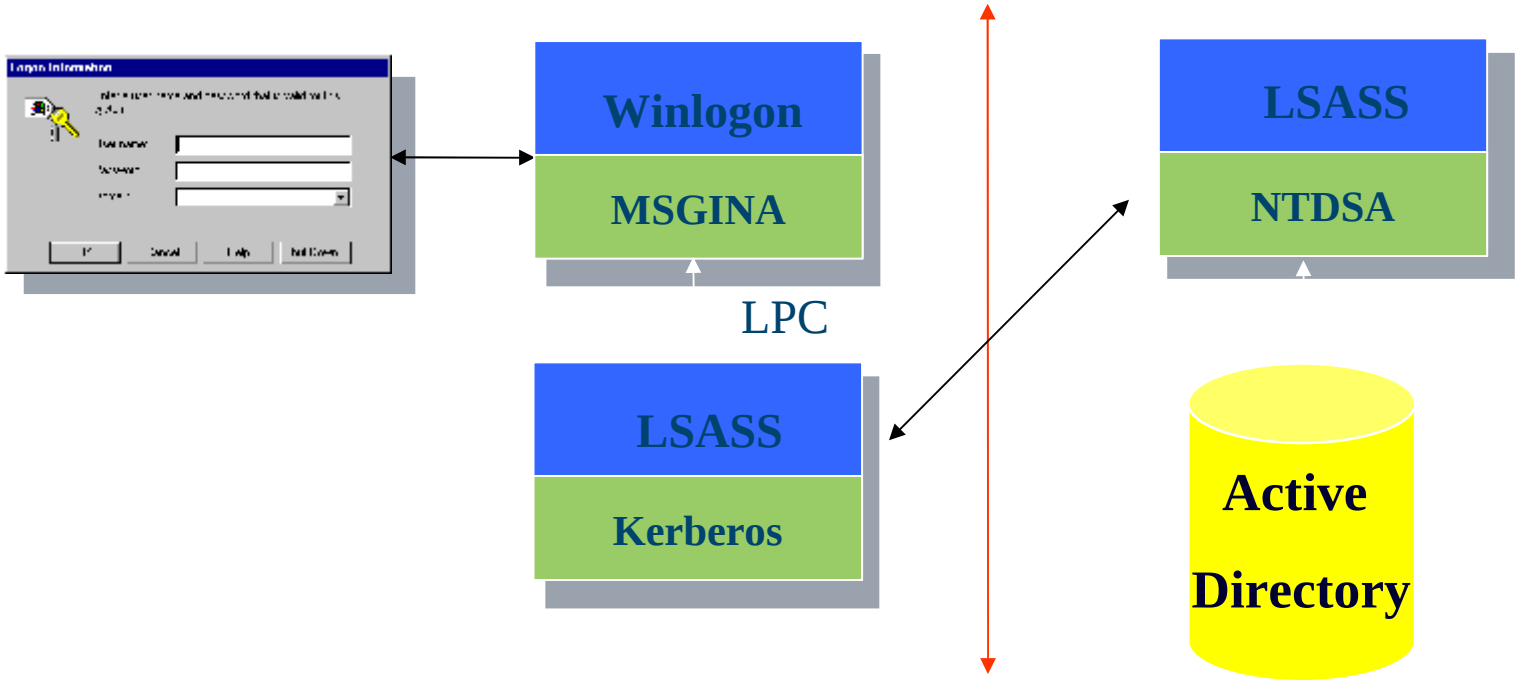


Local Logon



Remote Logon - Active Directory

- If the logon is for a domain account, the encrypted credentials are sent to LSASS on the domain controller:

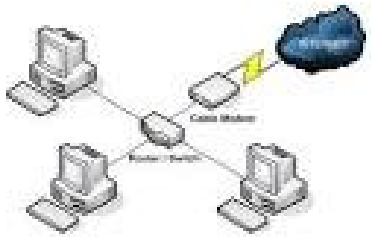


Local vs. Domain Accounts

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

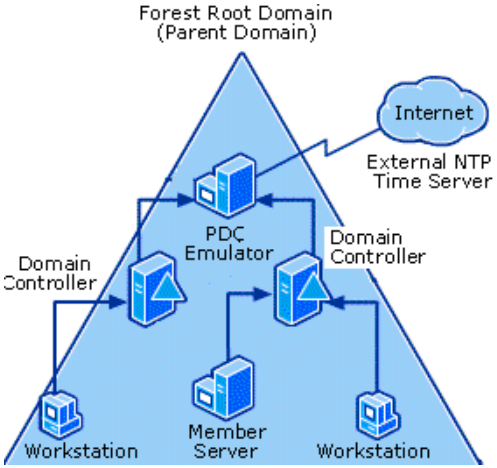
Workgroup Joined

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



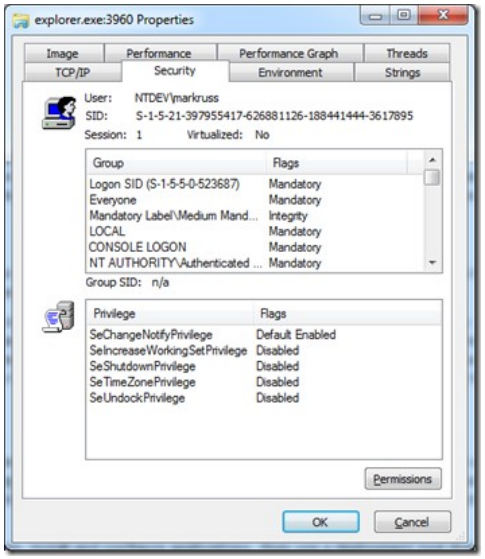
Domain Joined

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



User Management, Tokens and Access Control Lists

Windows Login Example



- Administrator creates a user account (full name, username, password, group, privileges)
- Windows creates an Security ID (SID)
  - 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

Security Identifiers - SIDs

- Windows uses Security Identifers (SIDs) to identify security principles:
  - Users, Groups of users, Computers and Domains

SIDs consist of: Example: S-1-5-21-3196711-41413171-350571-1000

- A revision level e.g. 1
- An identifier-authority value e.g. 5 (SECURITY_NT_AUTHORITY)
- One or more subauthority values, predifined

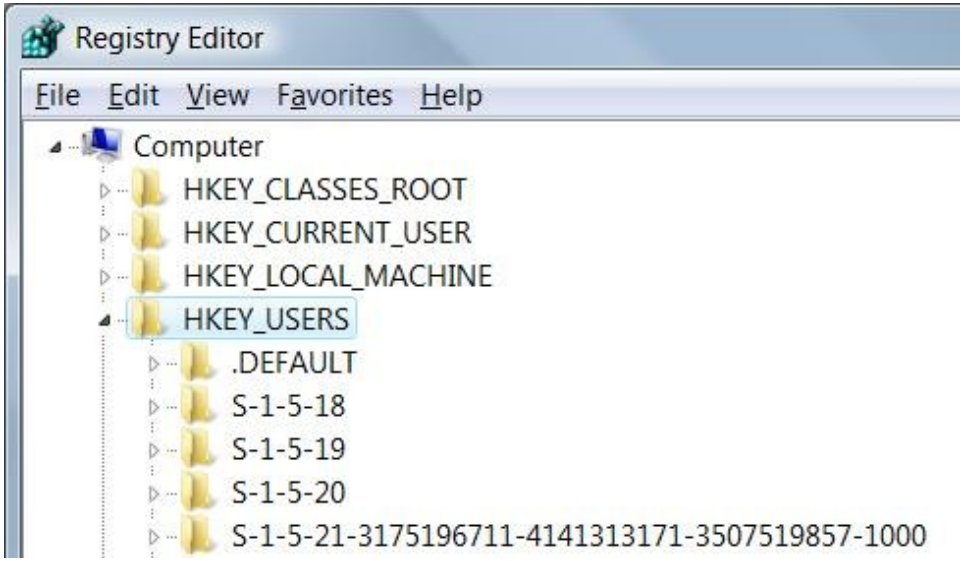
- SIDs are generally long enough to be globally statistically unique
- Setup assigns a computer an SID
- Users and groups on the local machine are assigned SIDs that are rooted with the computer SID, with a Relative Identifier (RID) at the end
  - Some local users and groups have pre-defined SIDs (eg. World = S-1-1-0)
  - RIDs start at 1000 (built-in account RIDs are pre-defined)

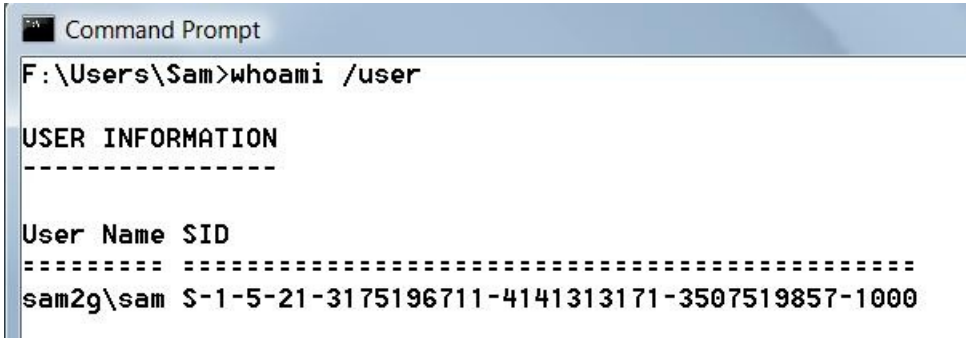
Security Identifiers (SIDs)

- User account

  has SID that uniquely

  identifies it





Windows Login Example
Assuming Active Directory

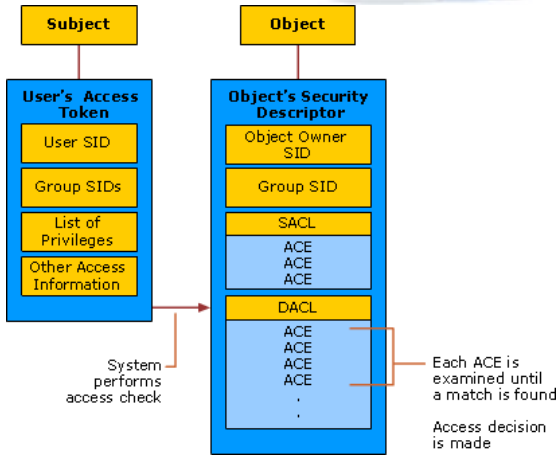
- User: **FredMgr**
- User logs in with keyboard
- Information is sent to the AD (domain controller)
- If successful,
  - User name and password or other authentication is good,
  - Token is generated and sent to user
- Token contains
  - User's SID
  - Group membership
  - Privileges

Windows Login Example

Security Token

Used ID:	FredMgr
Group Ids:	Users
	Mgrs
	Everyone
Privileges:	None

The Token



- Whats in a token?
  Contains a list of Security ID's associated with a user account
  - You can have multiple SIDs because you belong to multiple groups
  - So, when user tries to access a resource such as a file, token is used for access control
- How is token used?
  - Object, say a file will have an Access Control List (ACL) that specifies SID's permitted to access the object
  - If one of SID's in users token matches SID in Object's ACL, user granted access

Access Control List (ACL)

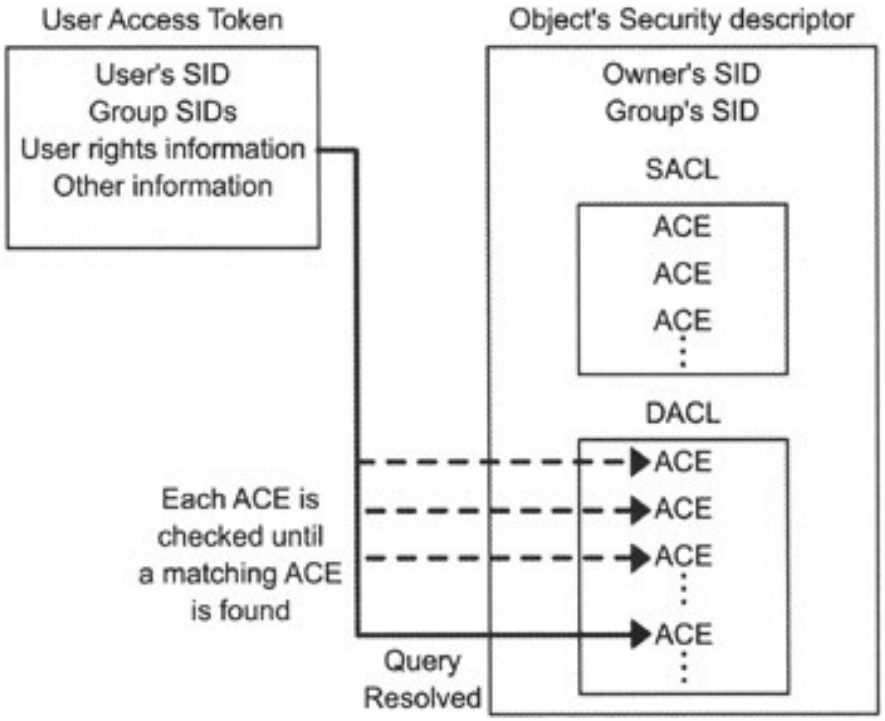
- Access to objects in Windows is through Discretinary ACL's
- Discretionary ACL
  - Grants or denies access to protected resources such as files, shared memory, etc.
- System ACL
  - Used for auditing and to enforce mandatory integrity policy (Vista)

Access Control Lists (ACL) (continued)

- Objects needing protection are assigned an ACL that includes
  - SID of object owner
  - List of access control entries (ACEs)
- Each ACE includes a SID and Access Mask
  - Access mask could include
    - Read, Write, Create, Delete, Modify, etc.

Access Control Example

- User opens text file



Protecting Objects

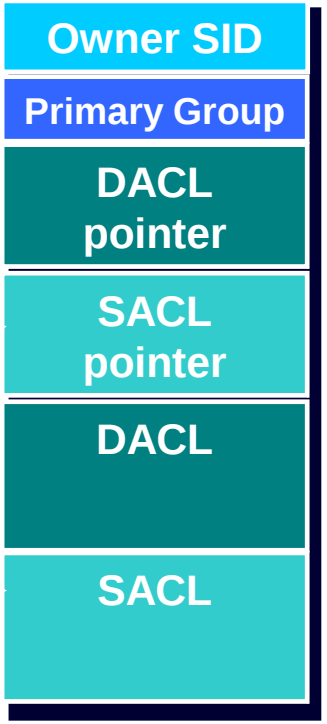
- Access to an object is through Security Reference Monitor (SRM),
  - Performs access validation at time object is opened by a process
- Access validation has following components:
  - Desired Access: Type of access
    - Must be specified up front,
    - Include all accesses performed on the object as a result of the validation.
  - Token: Identifies the User that owns the process,

    Plus user privileges

    The object's Security Descriptor
    - Contains a Discretionary Access Control List (DACL),
    - Describes the types of access to the object users are allowed.

Security Descriptors

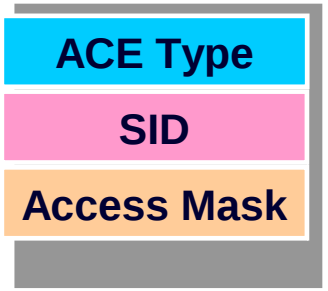
- Descriptors are associated with objects: e.g. files, Registry keys, application-defined
- Descriptors are variable length



Discretionary Access Control Lists DACLs

- DACLs consist of zero or more Access Control Entries
  - A security descriptor with no DACL allows all access
  - A security descriptor with an empty (0-entry) DACL denies everybody all access

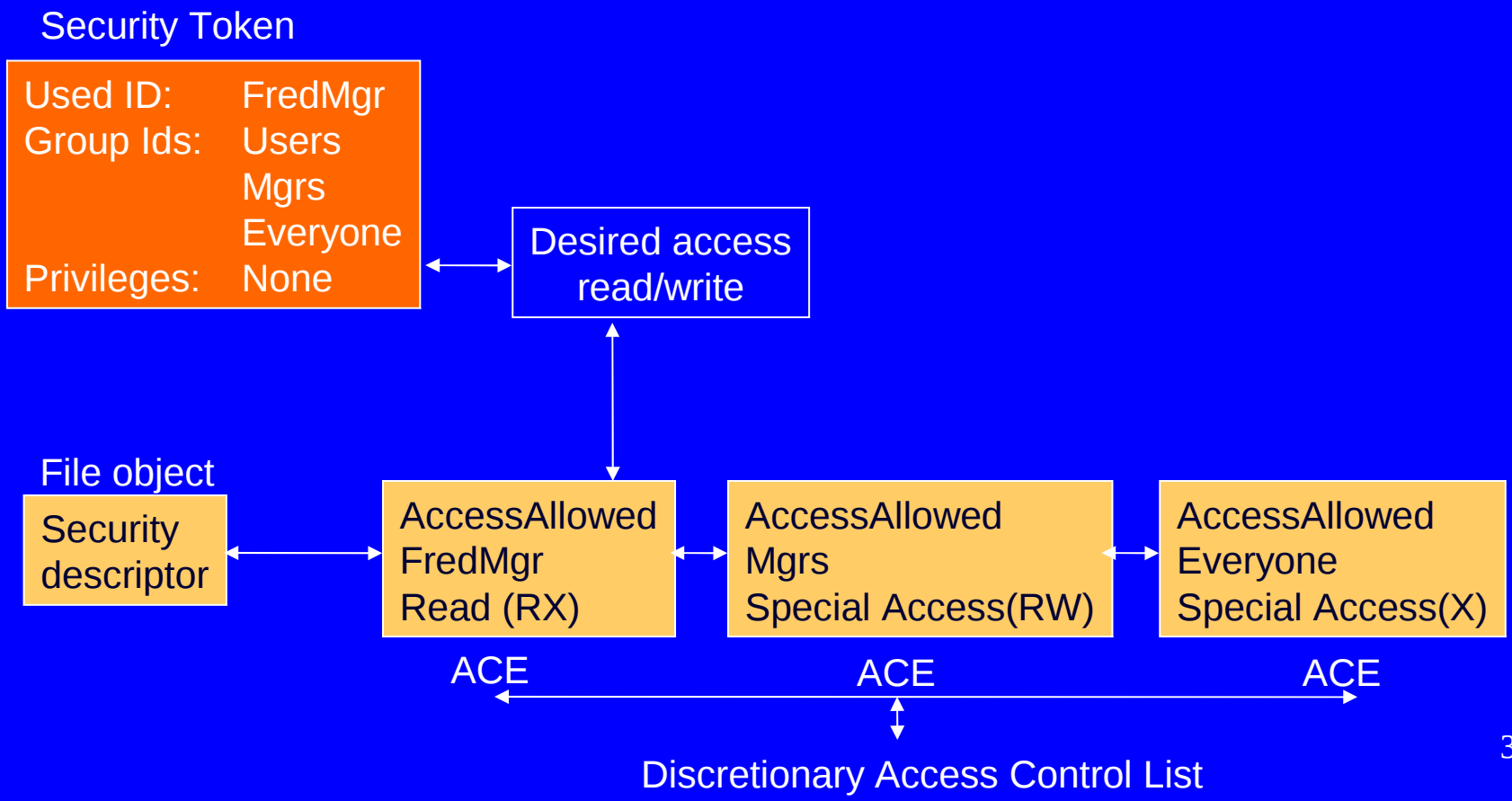
- An ACE is either “allow” or “deny”



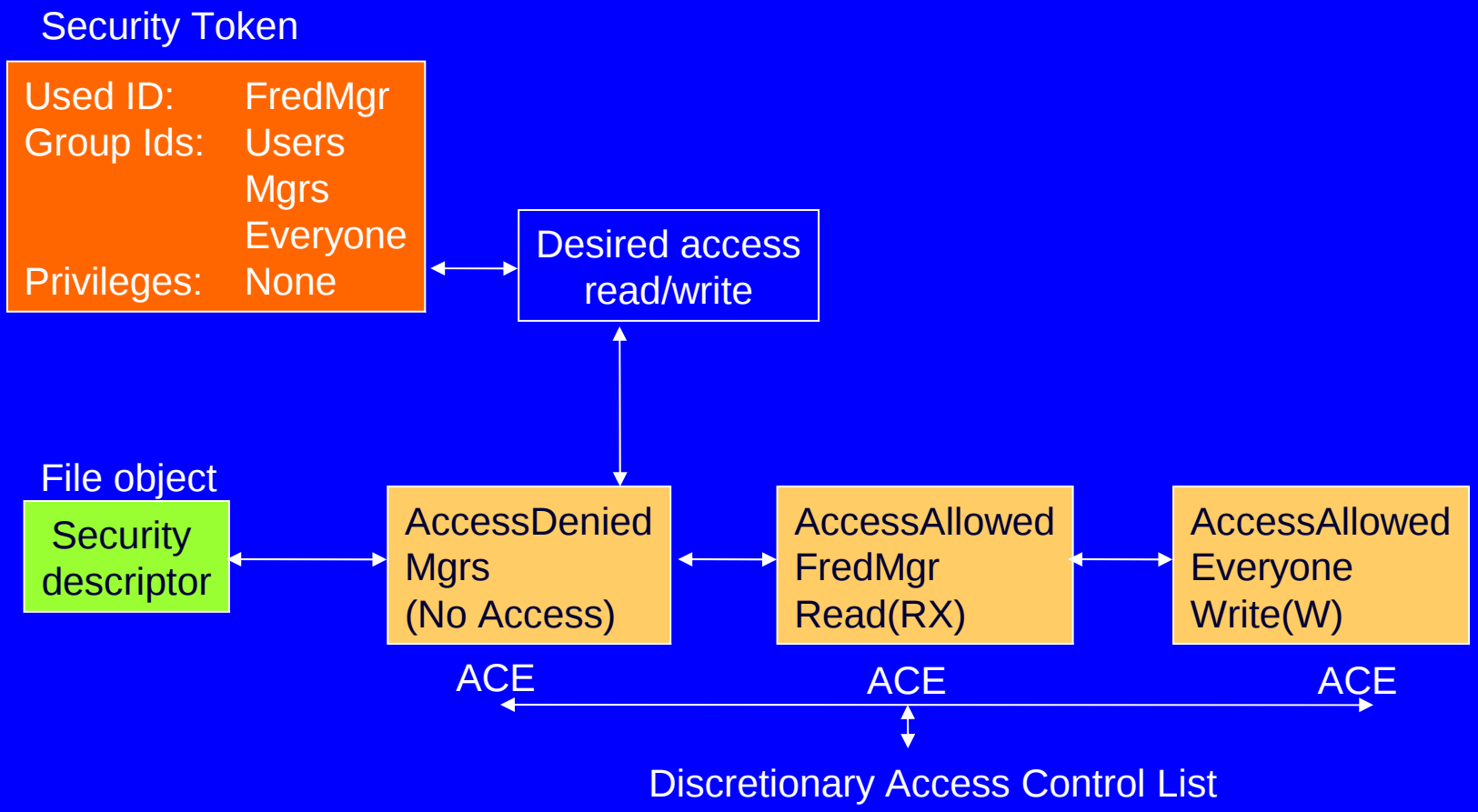
Access Check

- ACEs in the DACL are examined in order
  - Does the ACE have a SID matching a SID in the token?
  - If so, do any of the access bits match any remaining desired accesses?
  - If so, what type of ACE is it?
    - Deny: return ACCESS_DENIED
    - Allow: grant the specified accesses and if there are no remaining accesses to grant, return ACCESS_ALLOWED
  - If we get to the end of the DACL and there are remaining desired accesses from the user, return ACCESS_DENIED

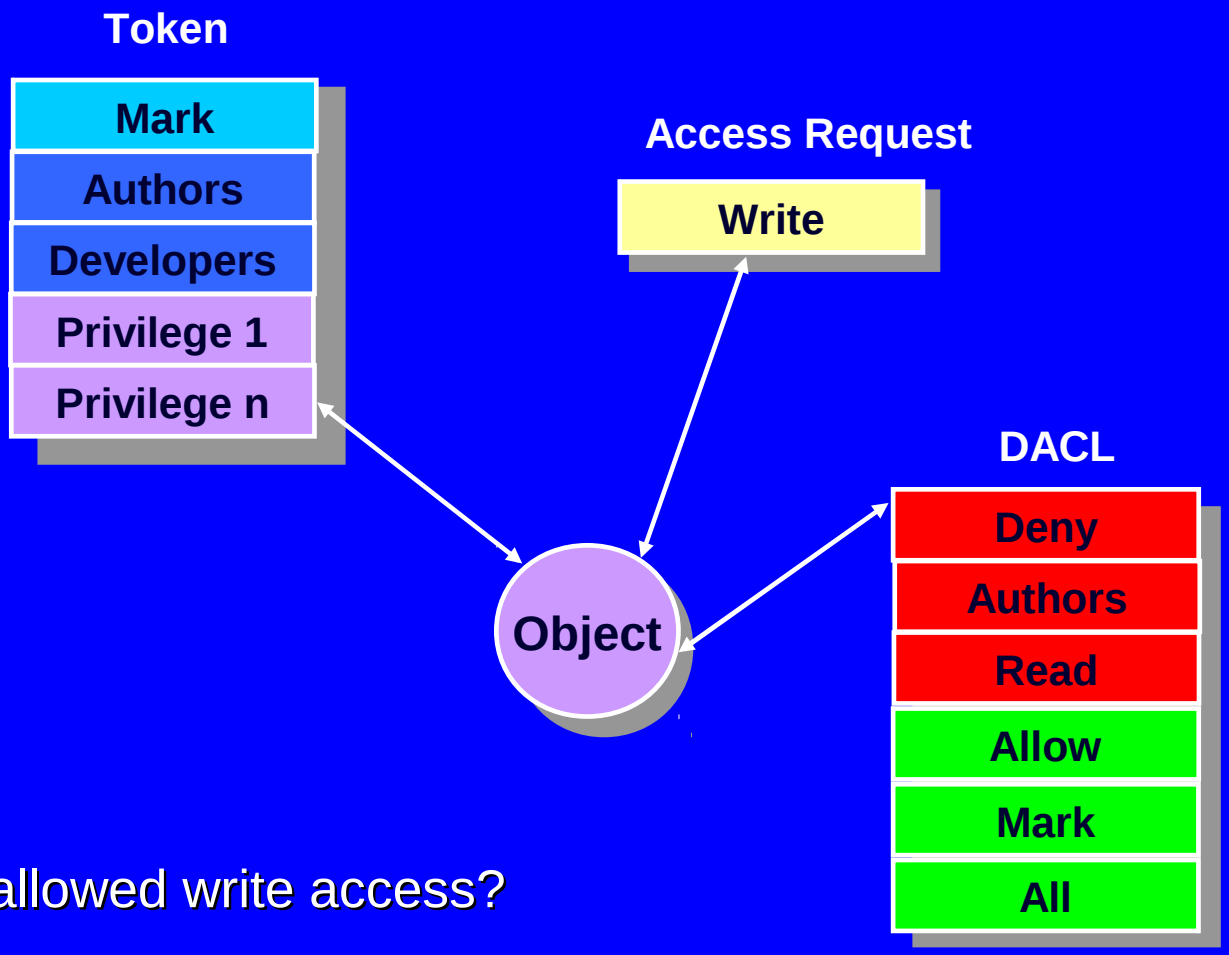
Example: Access granted



Example: Access denied



Access Check Quiz



Is Mark allowed write access?

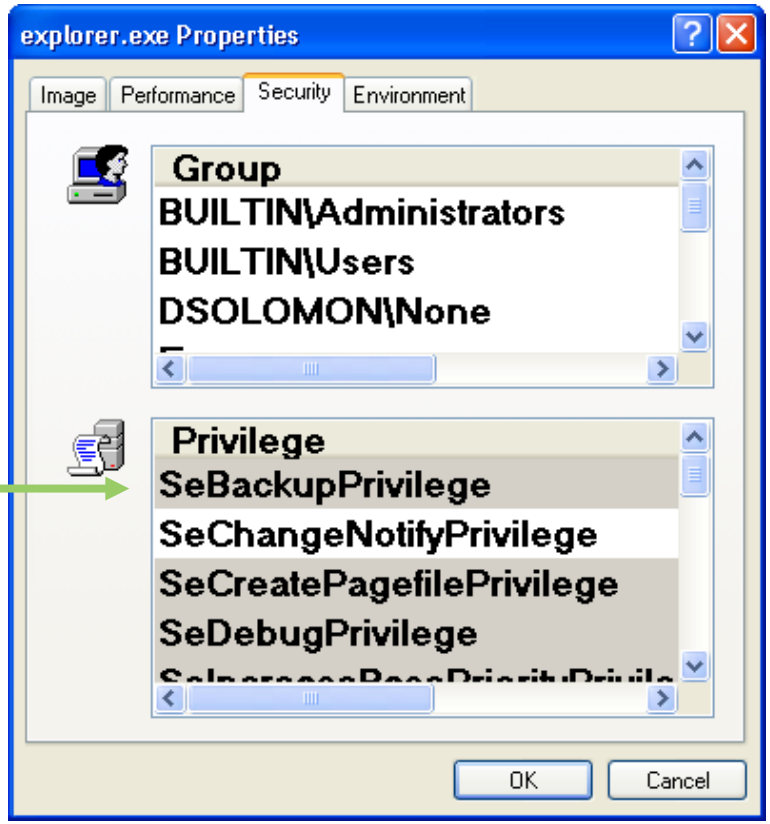
yes

Access Special Cases

- An object's owner can always open an object with WRITE and READ permission
- An account with "take ownership" privilege can claim ownership of any object
- An account with backup privilege can open any file for reading even if all others are denied, **Why?**
- An account with restore privilege can open any file for write access

Privileges

- Specify which system actions a process (or thread) can perform
- Privileges are associated with groups and user accounts
  - There are sets of pre-defined privileges associated with built-in groups (e.g. System, Administrators)
- Examples include:
  - Backup/Restore
  - Shutdown
  - Debug
  - Take ownership



Difference Between Privilege and Permission

- **Permisson**
  - Implies consent to user group to perform an action
  - Property of an object, r,w,x, modify
- **Privilege**
  - Permission given to user or group
  - Is a property of an agent / user and lets them do things which are not ordinarily allowed

Windows Privilege Levels

- Vista, Windows 7 on Up
  - Users have more choices
  - Can operate as a restricted normal user and still be able to do almost everything in OS
  - And when necessary, you can be elevated to an administrator for a short time to accomplish some task
  - Called UAC – User Account Control
    - Still alive and well in Windows 10

https://docs.microsoft.com/en-us/windows/access-protection/user-account-control/how-user-account-control-works

User Account Control

What does it do?

- UAC allows an administrator to enter credentials during a non-administrator's user session
- Perform occasional administrative tasks without having to switch users, log off, or use the Run as command
- UAC can also require administrators to specifically approve applications that will make "system-wide" changes before those applications are allowed to run

User Account Control

- Windows Vista and 7, how it works:
  - **Admin Approval Mode (AAM)**, by default, is not enabled for the Built-in Administrator Account in Windows Vista or 7
  - **Built-in Administrator Account** is disabled by default in Windows Vista, and first user account created is placed in local Administrators group, and AAM is enabled for that account

Benefits of UAC

- **What does it do for you?**
- **Admin Approval Mode** helps prevent malicious programs from silently installing without an administrator's knowledge
- It also helps protect from inadvertent system-wide changes
- Lastly, it can be used to enforce a higher level of compliance
  - Administrators must actively consent or provide credentials for each administrative process

Tasks Only Administrators Can Perform

- Create, change, and delete user accounts and groups
- Install and uninstall programs
- Configure automatic updating or install Windows updates manually
- Install an ActiveX control
- Install or remove hardware device drivers
- Share folders
- Set permissions
- Access all files, including those in another user's folder
- Take ownership of files
- Copy or move files into the %ProgramFiles% or %SystemRoot% folders
- Restore backed-up system files
- Grant rights to other user accounts and to themselves
- Configure Parental Controls
- Configure Windows Firewall

Tasks Available to Standard Users

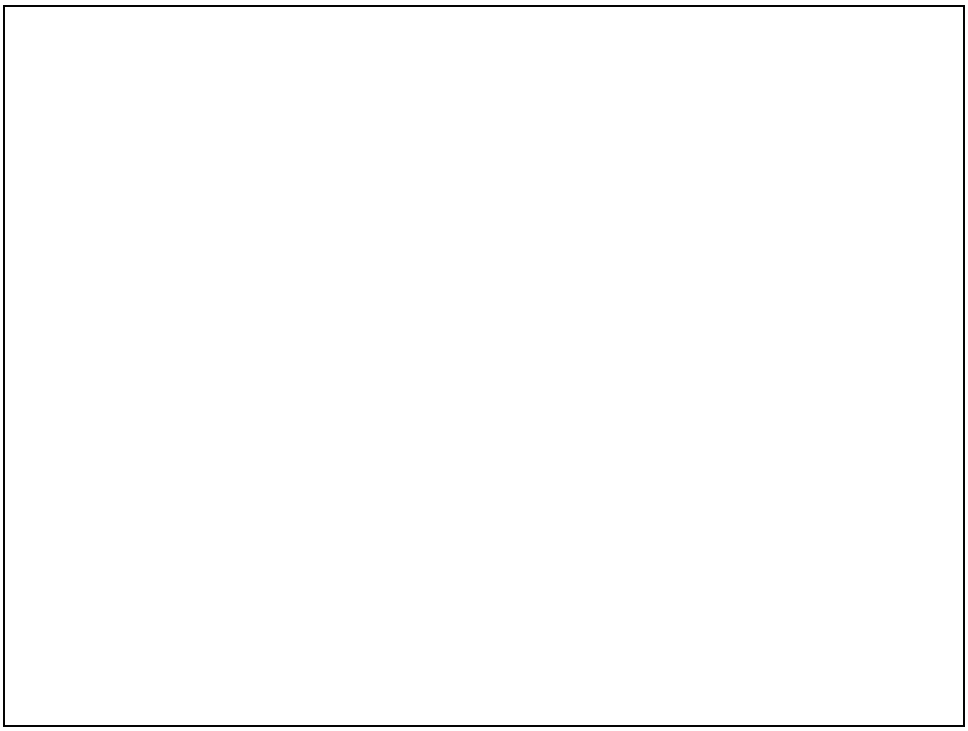
- Change the password and picture for their own user account
- Use programs that have been installed on the computer
- Install approved ActiveX controls
- Configure a secure Wi-Fi connection
- View permissions
- Create, change, and delete files in their document folders and in shared document folders
- Restore their own backed-up files
- View the system clock and calendar, and change the time zone
- Configure power options
- Log on in Safe Mode

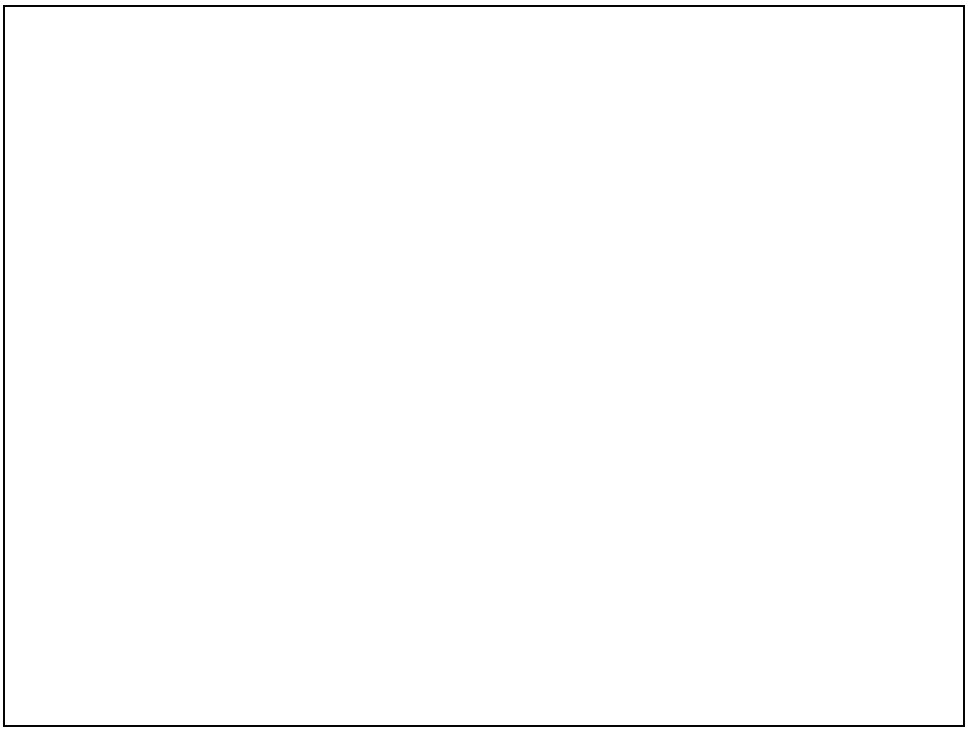
Summary

- Windows Operating Systems by design
  - Is genuinely trying to operate securely
- Users and processes have access to resources through security mechanisms
  - Mostly Discretionary Access control through their identities and group affiliations
- Want concept of **Least Privilege** to be in effect
- User Account Control assists with that in Windows
  - Like “sudo” in Linux
  - Helps with restricting access to system resources

The End

- Next Time: More Desktop
- Check for the lab
  - It will be a Kali Linux Install lab





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
- Review overall security architecture of Windows 2000 and later
  - Look at Security defenses built into Windows

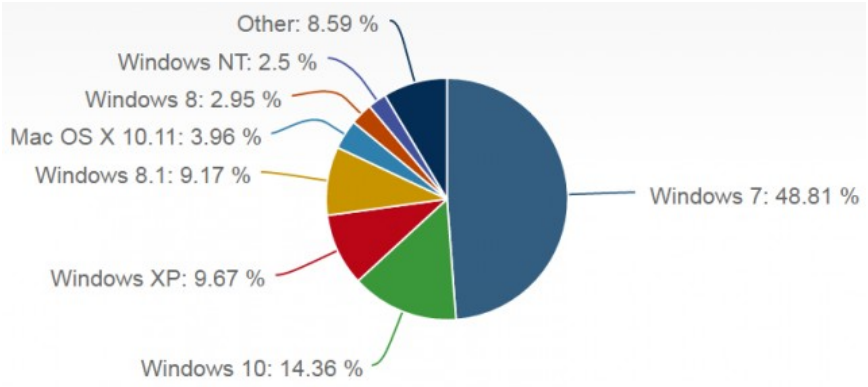
Windows is the world's most popular operating system and as such has a number of interesting security-related advantages and challenges. The major advantage is any security advancement made to Windows can protect hundreds of millions of nontechnical users, and advances in security technologies can be used by thousands of corporations to secure their assets. The challenges for Microsoft are many, including the fact that security vulnerabilities in Windows can affect millions of users. Of course, there is nothing unique about Windows having security vulnerabilities; all software products have security bugs. However, Windows is used by so many nontechnical users that Microsoft has some interesting engineering challenges. This chapter begins with a description the overall security architecture of Windows 2000 and later. It is important to point out that versions of Windows based on the Windows 95 code base,including Windows 98, Windows 98 SE, and Windows Me, had no real security model, in contrast to Windows NT and later versions. The Windows 9x codebase is no longer supported. The remainder of the chapter cover the security defenses built into Windows, most notably the new defenses in Windows Vista.

History

- In 1988, Microsoft created “new technology” (NT) portable operating system - supported both OS/2 and POSIX API's. NT supported servers and desktop workstations.
- During development NT was changed to use the Win32 API, reflecting the popularity of the Windows 3.0 Win16 API
- Windows XP was released in 2001 - replace earlier versions of Windows based on MS/DOS, such as Windows98 and Windows ME.
- Windows XP updated in 2005 to provide support AMD64 compatible CPUs, bringing support for 64-bit desktop systems.
- Windows Vista was released in late 2006, but was poorly received due to initial problems with application and device compatibility
- Windows 7 was released in late 2009, greatly improving on Vista.
- Windows 8 was released in October 2012. Not well received.

Windows 10 was first released on July 29, 2015. Windows 10 introduces "universal apps"; Apps can be designed to run across multiple Microsoft product families with nearly identical code

Operating System Market Share



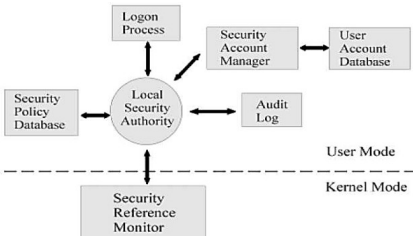
Background: Windows

- **Advantages**
  - User friendly
  - Enhancements can help millions of users
  - Defects found quickly because of widespread use

- **Disadvantages**
  - Security defects can leave millions vulnerable
  - Non-technical user-base
  - Industry dominance leaves MS handcuffed - any move to expand capabilities seen as anticompetitive
  - Forced to support legacy code and systems

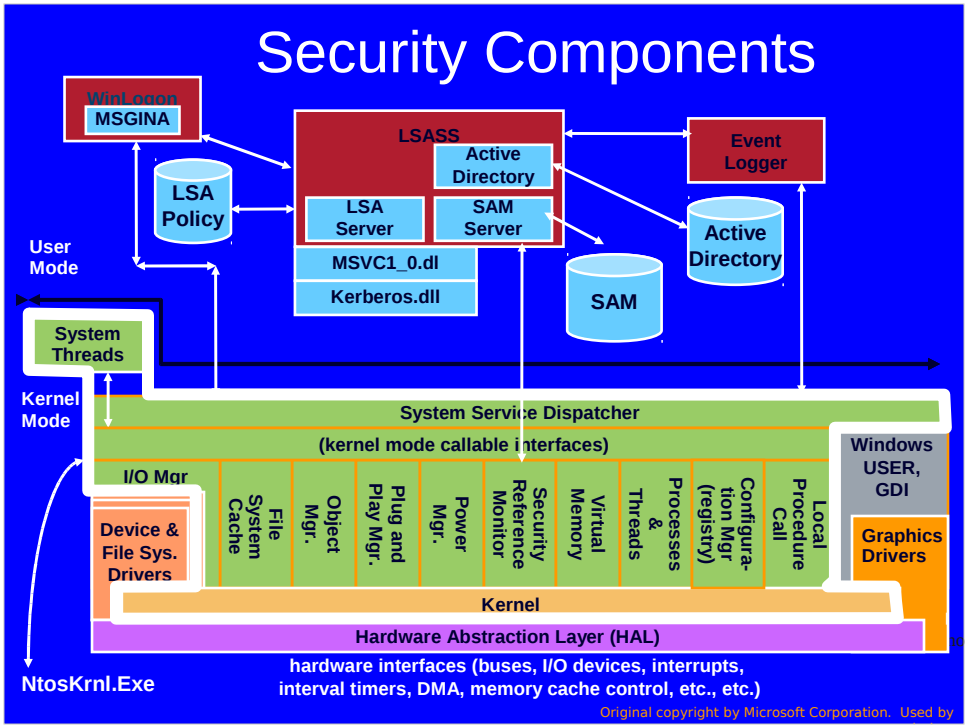
Windows Security Architecture

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



Anyone who wants to understand windows security must have knowledge of the basic fundamental security blocks of security system.

There are many components in windows that make up the fundamental security infrastructure.



Red boxes are security-related processes and blue are the security-related databases and DLLs.

Security Reference Monitor (SRM)

- Kernel Mode Component that
  - Performs Access Checks
  - Generates Audit Log Entries
  - Manipulates User Privileges
- Group of functions in Ntoskrnl.exe
  - Some functions documented in DDK (Windows Driver Devolopment Kit)
  - Exposed to user mode by Windows API calls

9

Its kernel mode component that performs access checks, generates audit log entities, and manipulates user privileges

Simply put it checks for proper authorization before granting access to objects

Object manager is client of SRM: It asks SRM if the process has the proper rights to execute a certain type of action on an object

Uses Access Control Lists to do this, which we will cover later in this presentation

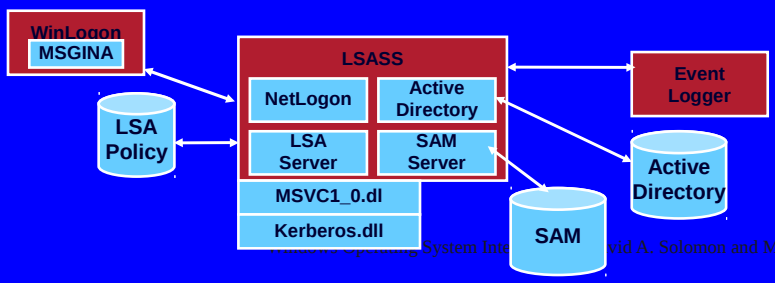
Implement auditing function to keep track of attempts to access an object

Implements DoD C2 level security: Department of Defense: C2 Level Security:

The following list includes some of the most important requirements of C2-level security, as defined by the U.S. Department of Defense:

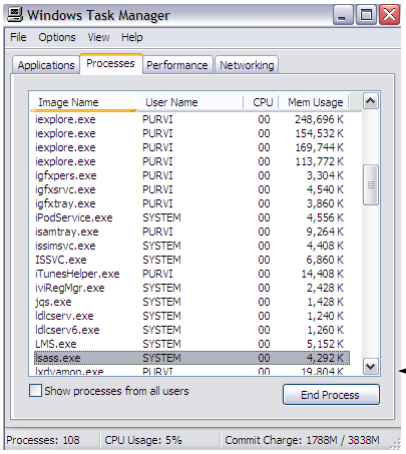
Security Components

- Local Security Authority
  - User-mode process (\Windows\System32\Lsass.exe) that implements policies (e.g. password, logon), authentication, and sending audit records to the security event log
  - LSASS policy database: registry key HKLM\SECURITY



Red boxes are security-related processes and blue are the security-related databases and DLLs.

Local Security Authority (LSA)



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

← Continuously runs as a process

Resides in user-mode process named lsass.exe and is responsible for enforcing local security policy in windows.

Responsible for validating users for both local and remote logon

Issues security tokens to accounts as they log on to the system

Security policy includes

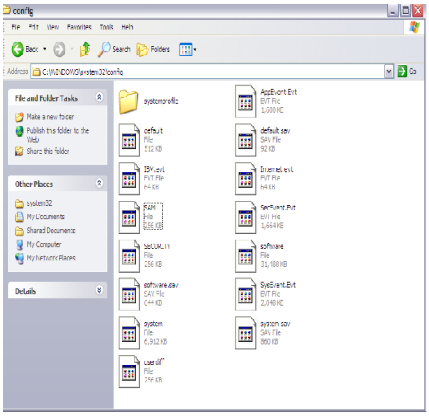
Password policy (complexity rules …expiration times etc.)

Auditing policy and privilege settings

During the local (interactive) logon to a machine, a person enters their name and password to the logon dialog. This information is passed to the LSA, which then calls the appropriate authentication package. The password is sent in a nonreversible secret key format using a one-way hash function. The LSA then queries the SAM database for the user's account information. If the key provided matches the one in the SAM, the SAM returns the users SID and the SIDs of any groups the user belongs to. The LSA then uses these SIDs to generate the security access token.

Was successfully exploited in the Sasser worm (by an 18 year old German student) :2004

LSASS Components
Security Account Manager (SAM)



- A database that stores user accounts and local users and groups security information
- SamSrv.exe

Database that stores user accounts and relevant security information about local users and local groups

Accounts Manager (SAM) is a registry file

Stores users' passwords in a hashed format

When a user logs on to a computer using a local account, the SAM process (Samsrv) takes the logon information and performs a lookup against the SAM database, which resides in the windows system32/config directory(Something similar in UNIX, think etc/password). If credential match, then the user can log on to the system, assuming there are no other factors preventing logon, such as logon time restrictions or privilege issues.

Note that SAM does not perform the logon; that is the job of the LSA.

The SAM file is binary rather than text and passwords are stored using the MD4 hash algorithms.

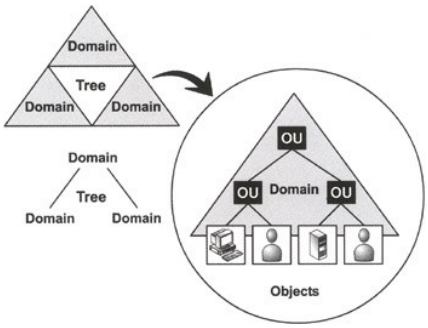
On windows vista, the SAM stores password information using a password-based key derivation function (PBKCS).

In an attempt to improve the security of the SAM database against offline software cracking, Microsoft introduced the SYSKEY function in Windows NT 4.0. When SYSKEY is enabled, the on-disk copy of the SAM file is partially encrypted, so that the password hash values for all local accounts stored in the SAM are encrypted with a key (usually also referred to as the "SYSKEY").

In the case of online attacks, it is not possible to simply copy the SAM file to another location. The SAM file cannot be moved or copied while Windows is running, since the Windows kernel obtains and keeps an exclusive filesystem lock on the SAM file, and will not release that lock until the operating system has shut down or a blue screen exception has been thrown. However, the in-memory copy of the contents of the SAM can be dumped using various techniques, making the password hashes available for offline brute-force attack.

LSASS Components
Active Directory

- Directory Service
  - Server-based authentication, non-local
  - Centrally managed by a domain controller



13

A directory service used to store information about the network resources across a domain and also centralizes the network.

Server based authentication

Centrally managed

Its microsofts's LDAP directory included with window server 2000 and later.

All client versions of windows, including windows XP and windows Vista, can communicate with AD to perform security operations including account logon

Windows client will authenticate using AD when the user logs on to the computer using a domain account rather than a local account.

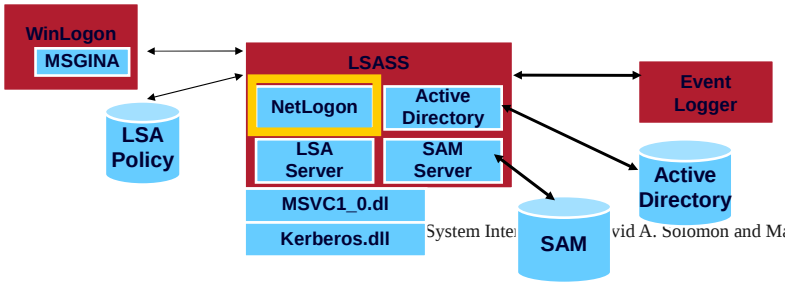
Like the SAM scenario, the user's credential information sent securely across the network, verified by AD, and then if the information is correct, the user can Logon.

LSASS Components

- Active Directory
  - A directory service contains database that stores information about objects in a domain
  - A *domain* is a collection of computers and their associated security groups that are managed as a single entity
  - Active Directory server, implemented as a service, \Windows\System32\Ntdsa.dll, that runs in the Lsass process
- Authentication packages
  - DLLs that run in context of Lsass process and that implement Windows authentication policy:
    - LanMan: \Windows\System32\Msvc1_0.dll
    - Kerberos: \Windows\System32\Kerberos.dll
    - Negotiate: uses LanMan for local machine or Kerberos, for domain machine

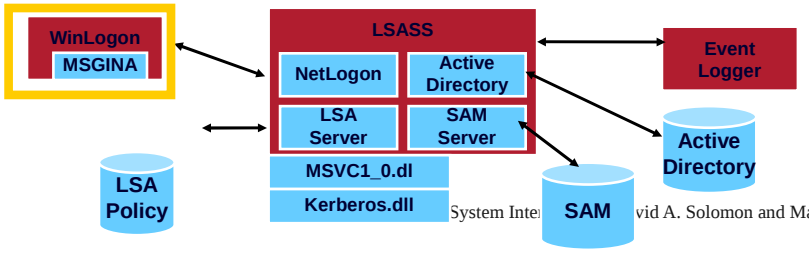
LSASS Components

- **Net Logon Service (Netlogon)**
  - A Windows service (\Windows\System32\Netlogon.dll) that runs inside Lsass and responds to Microsoft LAN Manager 2 Windows NT (pre-Windows 2000) network logon requests
  - Authentication is handled as local logons are, by sending them to Lsass for verification
  - Netlogon also has a locator service built into it for locating domain controllers

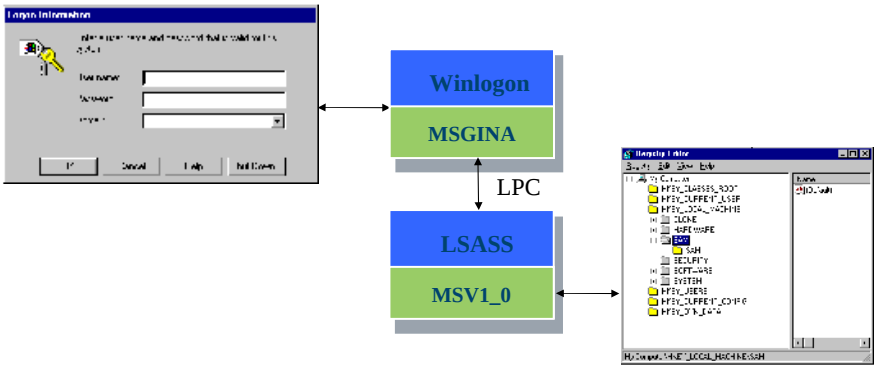


Security Components

- **Logon process (Winlogon)**
  - A user-mode process running \Windows\System32\Winlogon.exe that is responsible for responding to the LSASS and for managing interactive logon sessions
  - Graphical Identification and Authentication (GINA)
  - A user-mode DLL that runs in the Winlogon process and that Winlogon uses to obtain a user's name and password or smart card PIN - Default is \Windows\System32\Msgina.dll

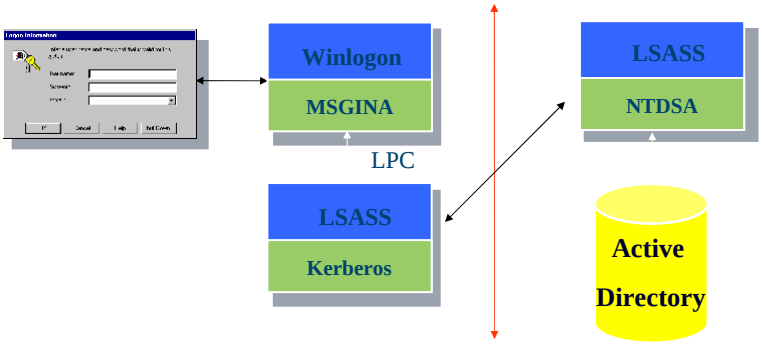


Local Logon



Remote Logon - Active Directory

- If the logon is for a domain account, the encrypted credentials are sent to LSASS on the domain controller:



Local vs. Domain Accounts

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

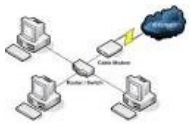
Networked windows computer can be one of two configuration either domain joined or workgroup

Difference between a workgroup and a domain is simply where accounts are authenticated

The workgroup has no domain controllers; authentication is performed on each computer, and a domain authenticates accounts at domain controllers running AD

Workgroup Joined

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

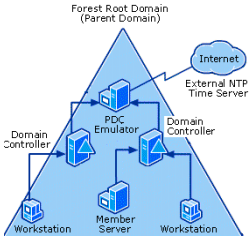


Wokgroup: when computer is workgroup, only local accounts can be used, held in the SAM

Notion of workgroup smipy a collection of computers connected to one another using a network; but rather using central database of accounts in AD, the machines use only local accounts

Domain Joined

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



Domain joined: user can gain access to the computer using domain accounts, which are centrally managed in active directory

If they wish also can log on using local accounts but local accounts may not have access to domain resources such as networked printers, web servers, email servers and so on

Pros and Cons to each scenario

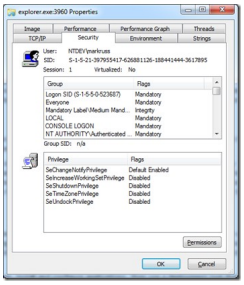
Domain has major advantage of being centrally managed; as such is much more secure

If environment has 1000 computers and an employee leaves, the user's account can be disabled centrally rather than on 1000 individual computer

Only advantage of using local accounts is that a computer does not need the infrastructure required to support a domain using AD

User Management, Tokens and Access Control Lists

Windows Login Example



- Administrator creates a user account (full name, username, password, group, privileges)
- Windows creates an Security ID (SID)
  - 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

23

now we know the basic elements that makes up the core windows security infrastructure. Lets go over example when user logs on to a windows system

before a user log on to windows network, domain administrator must add user's account information to the system; this will include username, account name(must be unique), password

optionally the admin can grant group membership and privileges

Windows creates an SID in the form of

S-1-5-21-AAA-BBB-CCC-RRR

S : means SID

1: SID version number

5: identifier authority

21: means not unique , which just means there is no guarantee of uniqueness, a SID is unique within a domain

AAA-BBB-CCC: unique number representing domain

RRR: called relative ID (RID) ; its unique number that increments by 1 as each SID unique

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

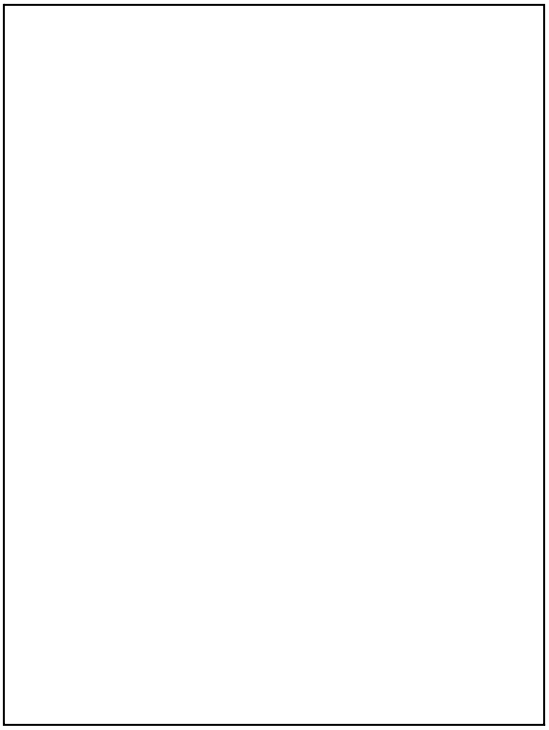
Security Identifiers - SIDs

- Windows uses Security Identifers (SIDs) to identify security principles:
  - Users, Groups of users, Computers and Domains

SIDs consist of: Example: S-1-5-21-3196711-41413171-350571-1000

- A revision level e.g. 1
- An identifier-authority value e.g. 5 (SECURITY_NT_AUTHORITY)
- One or more subauthority values, predifined

- SIDs are generally long enough to be globally statistically unique
- Setup assigns a computer an SID
- Users and groups on the local machine are assigned SIDs that are rooted with the computer SID, with a Relative Identifier (RID) at the end
  - Some local users and groups have pre-defined SIDs (eg. World = S-1-1-0)
  - RIDs start at 1000 (built-in account RIDs are pre-defined)



Windows Login Example
Assuming Active Directory

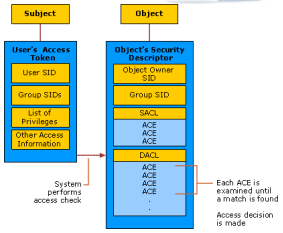
- User: **FredMgr**
- User logs in with keyboard
- Information is sent to the AD (domain controller)
- If successful,
  - User name and password or other authentication is good,
  - Token is generated and sent to user
- Token contains
  - User's SID
  - Group membership
  - Privileges

Windows Login Example

Security Token

Used ID:	FredMgr
Group Ids:	Users
	Mgrs
	Everyone
Privileges:	None

The Token



- Whats in a token?
  Contains a list of Security ID's associated with a user account
  - You can have multiple SIDs because you belong to multiple groups
  - So, when user tries to access a resource such as a file, token is used for access control
- How is token used?
  - Object, say a file will have an Access Control List (ACL) that specifies SID's permitted to access the object
  - If one of SID's in users token matches SID in Object's ACL, user granted access

Access Control List (ACL)

- Access to objects in Windows is through Discretinary ACL's
- Discretionary ACL
  - Grants or denies access to protected resources such as files, shared memory, etc.
- System ACL
  - Used for auditing and to enforce mandatory integrity policy (Vista)

An ACL is a list of permissions attached to an object

The list specifies who or what is allowed to access the object and what operations are allowed to be performed on the object.

Windows has two forms of access control list

Discretionary ACL (DACL):

Grants and denies access to protected resources in windows such as files, shared memory

System ACL (SACL)

Used for auditing and in windows vista used to enforce mandatory integrity policy

Objects that requires protection are assigned a DACL (and possible SACL), which includes SID of object owner as well as a list of access control entries (ACEs)

Each ACE includes a SID and an access mask

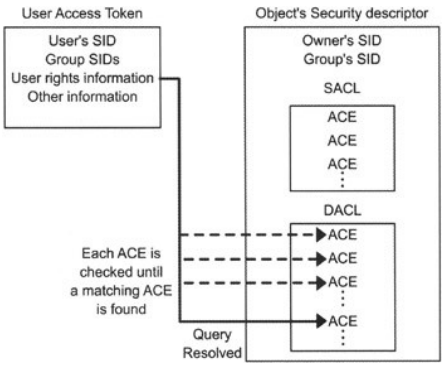
Access mask could include an ability to read, write, create, delete , modify and so on

Access Control Lists (ACL) (continued)

- Objects needing protection are assigned an ACL that includes
  - SID of object owner
  - List of access control entries (ACEs)
- Each ACE includes a SID and Access Mask
  - Access mask could include
    - Read, Write, Create, Delete, Modify, etc.

Access Control Example

- User opens text file



Protecting Objects

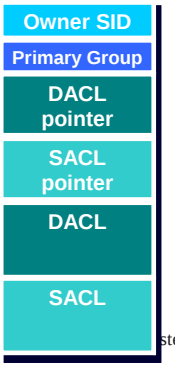
- Access to an object is through Security Reference Monitor (SRM),
  - Performs access validation at time object is opened by a process
- Access validation has following components:
  - Desired Access: Type of access
    - Must be specified up front,
    - Include all accesses performed on the object as a result of the validation.
  - Token: Identifies the User that owns the process,

    Plus user privileges

  The object's Security Descriptor
    - Contains a Discretionary Access Control List (DACL),
    - Describes the types of access to the object users are allowed.

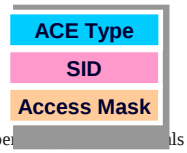
Security Descriptors

- Descriptors are associated with objects: e.g. files, Registry keys, application-defined
- Descriptors are variable length



Discretionary Access Control Lists DACLs

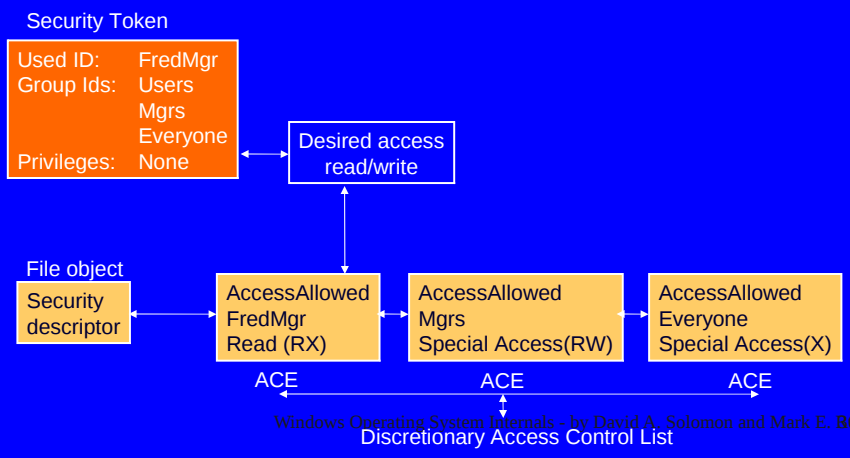
- DACLs consist of zero or more Access Control Entries
  - A security descriptor with no DACL allows all access
  - A security descriptor with an empty (0-entry) DACL denies everybody all access
- An ACE is either “allow” or “deny”



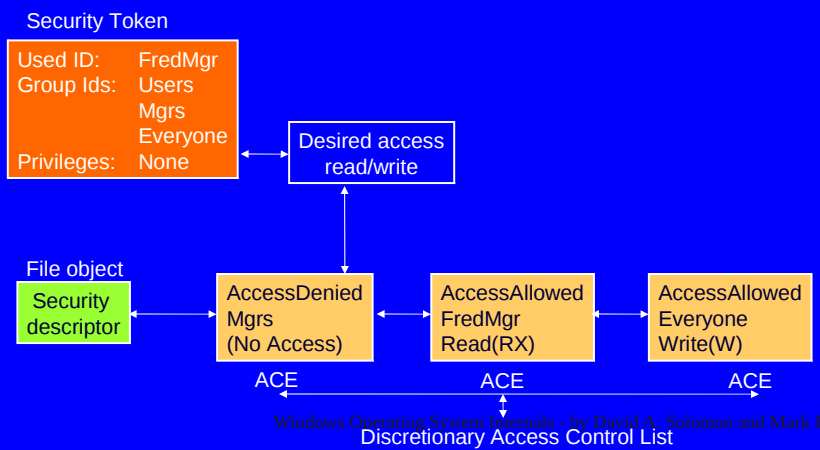
Access Check

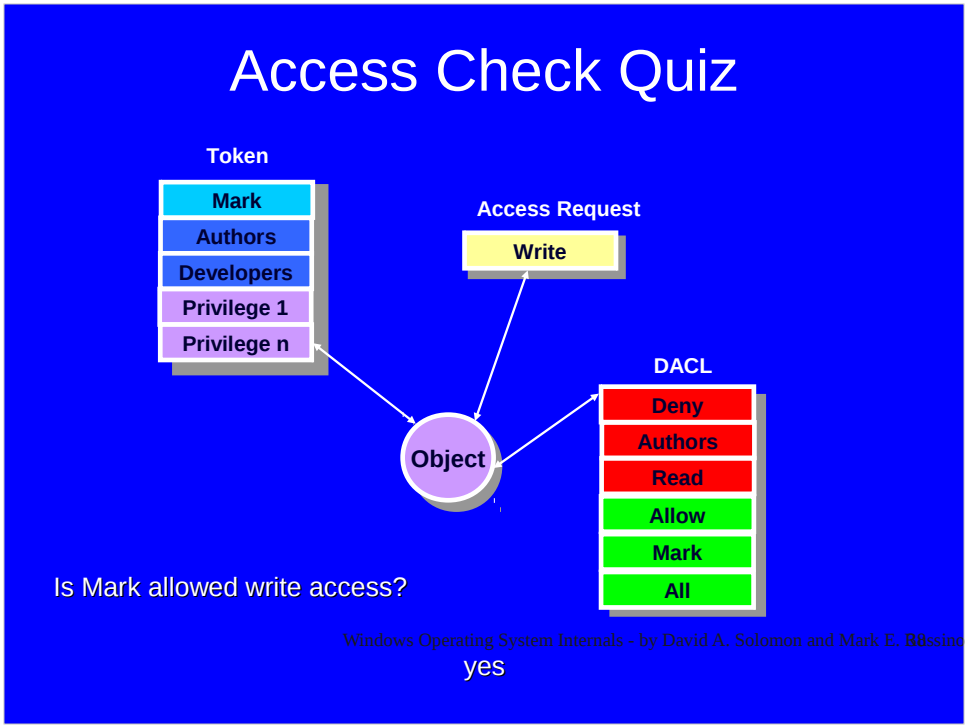
- ACEs in the DACL are examined in order
  - Does the ACE have a SID matching a SID in the token?
  - If so, do any of the access bits match any remaining desired accesses?
  - If so, what type of ACE is it?
    - Deny: return ACCESS_DENIED
    - Allow: grant the specified accesses and if there are no remaining accesses to grant, return ACCESS_ALLOWED
  - If we get to the end of the DACL and there are remaining desired accesses from the user, return ACCESS_DENIED

Example: Access granted



Example: Access denied





The first ACE does not apply to the access check since Mark is requesting WRITE access, but the ACE controls READ access. The second ACE does apply since WRITE is a subset of ALL.

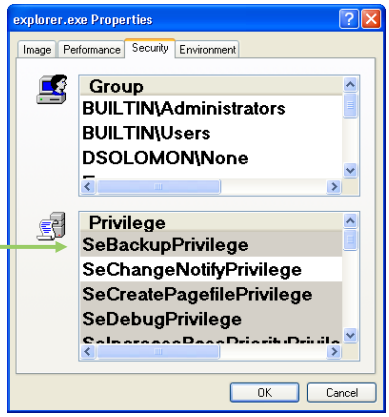
Note that WRITE does not imply READ. The security system does not understand the semantics of different access types, just their bit representations.

Access Special Cases

- An object's owner can always open an object with WRITE and READ permission
- An account with "take ownership" privilege can claim ownership of any object
- An account with backup privilege can open any file for reading even if all others are denied, **Why?**
- An account with restore privilege can open any file for write access

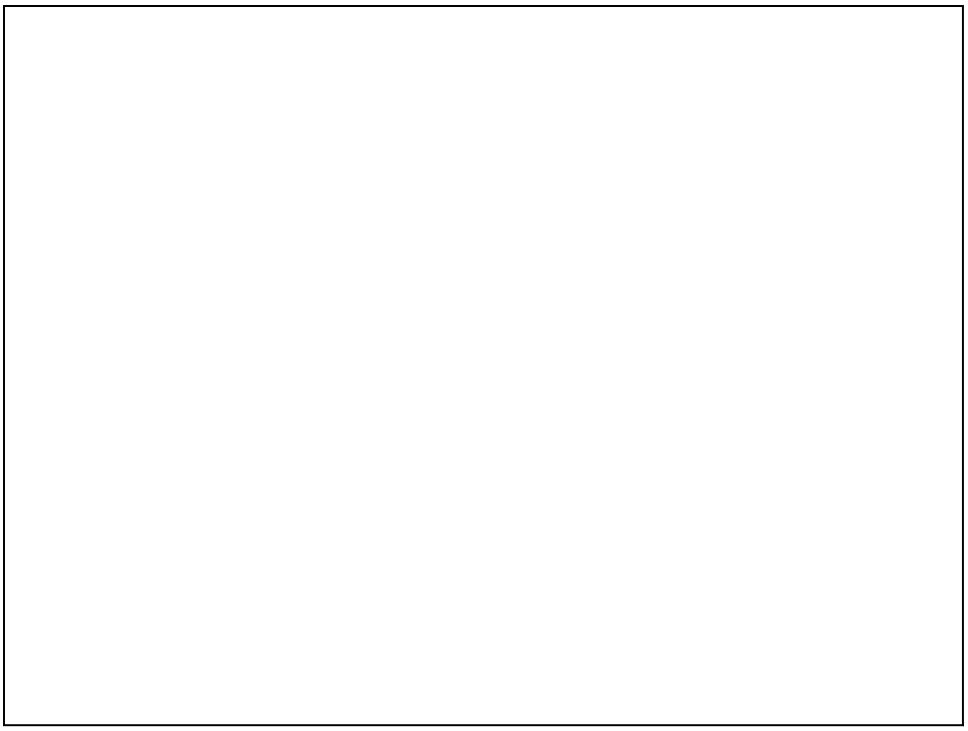
Privileges

- Specify which system actions a process (or thread) can perform
- Privileges are associated with groups and user accounts
  - There are sets of pre-defined privileges associated with built-in groups (e.g. System, Administrators)
- Examples include:
  - Backup/Restore
  - Shutdown
  - Debug
  - Take ownership



Difference Between Privilege and Permission

- **Permisson**
  - Implies consent to user group to perform an action
  - Property of an object, r,w,x, modify
- **Privilege**
  - Permission given to user or group
  - Is a property of an agent / user and lets them do things which are not ordinarily allowed



User Account Control

What does it do?

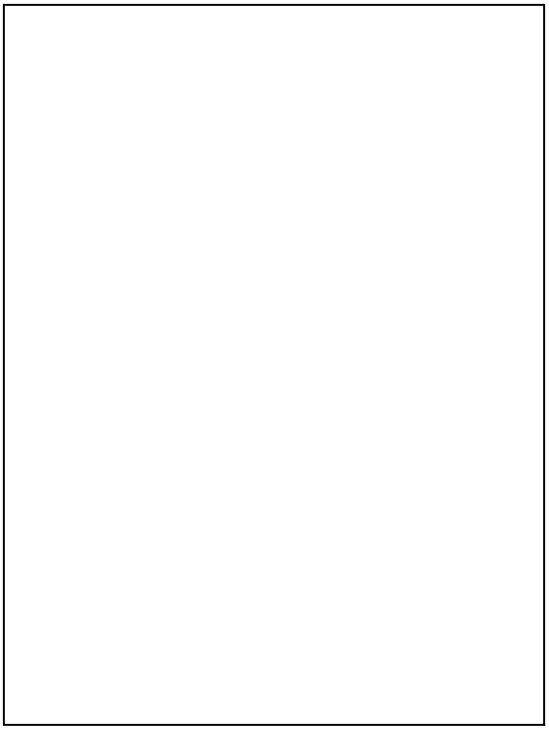
- UAC allows an administrator to enter credentials during a non-administrator's user session
- Perform occasional administrative tasks without having to switch users, log off, or use the Run as command
- UAC can also require administrators to specifically approve applications that will make "system-wide" changes before those applications are allowed to run

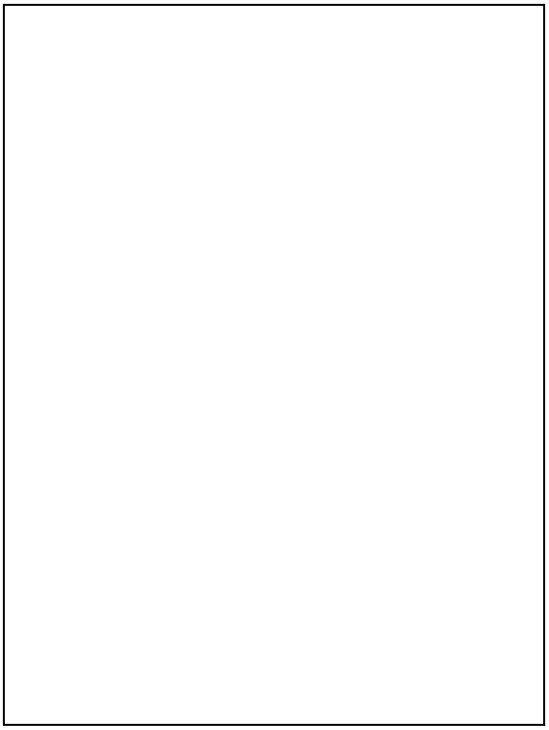
User Account Control

- Windows Vista and 7, how it works:
  - **Admin Approval Mode (AAM)**, by default, is not enabled for the Built-in Administrator Account in Windows Vista or 7
  - **Built-in Administrator Account** is disabled by default in Windows Vista, and first user account created is placed in local Administrators group, and AAM is enabled for that account

Benefits of UAC

- **What does it do for you?**
- **Admin Approval Mode** helps prevent malicious programs from silently installing without an administrator's knowledge
- It also helps protect from inadvertent system-wide changes
- Lastly, it can be used to enforce a higher level of compliance
  - Administrators must actively consent or provide credentials for each administrative process





Summary

- Windows Operating Systems by design
  - Is genuinely trying to operate securely
- Users and processes have access to resources through security mechanisms
  - Mostly Discretionary Access control through their identities and group affiliations
- Want concept of **Least Privilege** to be in effect
- User Account Control assists with that in Windows
  - Like “sudo” in Linux
  - Helps with restricting access to system resources