Windows Security

- Windows XP evolved from Windows 2000
- Windows 10, 8, 7 and Vista evolved from XP
- Similar security solution
  - Things have been added, but ideas are the same

- Standalone computers administered locally
- Domains used for centralized administration
  - Domain controller (DC) has information about users
    - Acts as a trusted third party

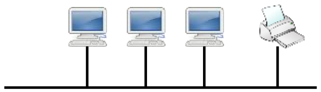
Rich support for managing security

Design Motivation

- Security was designed to meet requirements for C2 rating in Orange Book
  - **Secure logon** – users must be uniquely identified
  - **Discretionary access control** – Owner determines access
  - **Auditing** – Record security related events in a logfile
  - **Object reuse protection** – Initialize all objects before giving access to users
  - **Trusted path** – Functionality to detect spoofing attacks at authentication time (called SAS in Windows)
  - **Trusted facility management** – Separate accounts for users and administrators
- Windows NT 3.5 SP3 was the first Windows version to earn C2 rating (1995)
  - Windows NT 4 SP6a earned C2 rating in 1999

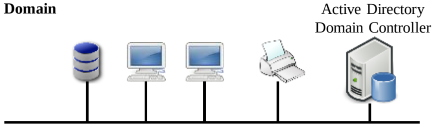
Networked Computers, Domains and Workgroups

Workgroup



- All accounts are local
- Resources can be shared but are managed locally
- Peer-to-peer
- No computer in charge
- Typically at most 20 computers

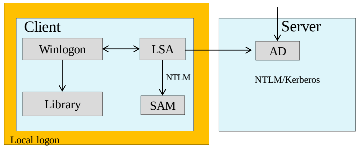
Domain



- Both local and domain accounts
- Login to domain account from any computer
- Centralized administration
- Can be thousands of computers

Windows Logon (somewhat simplified)

- Winlogon.exe handles the logon and responds to the Secure Attention Sequence (SAS)
  - CTRL+ALT+DEL
- Winlogon uses libraries that authenticates the user
  - Can be libraries for passwords, smartcards, biometric data etc
- Local Security Authority (LSA) creates an access token
  - LSA is responsible for the local security policy (who can log in, password policies, privileges, what should be audited etc)
- Password hashes are stored in SAM
  - Security Accounts Manager



Security Accounts Manager (SAM)

- Stores user account information
  - Username
  - Full name
  - Expiration date
  - Password dates (date of last change, expiry, when it can be changed next time, if it can be changed)
  - Logon hours and workstations (thrown out a certain time or continue)
  - Profile path and logon script name
  - Home directory
  - Groups
- Locked while machine is running

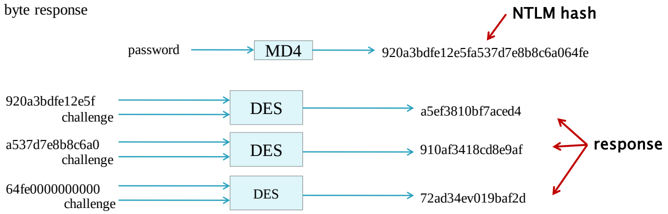
Local Accounts VS Domain accounts

- Local accounts
  - NTLM used as authentication protocol
- Domain accounts
  - Kerberos V5 used as authentication protocol
    - Mutual authentication
    - This will be covered in detail later in the course
  - NTLM used in some cases
    - Unilateral authentication

NTLM Hash and Protocol

Can you find problems here?

- Challenge response
- Server sends 8 byte random challenge
- Response calculated as:
  - MD4(password) gives 16 byte result (NTLM hash stored in SAM database)
  - Pad with 5 zero bytes → 21 bytes
  - Split into 3 DES keys and encrypt challenge with each key
  - 24 byte response



NTLM hash

password → MD4 → 920a3bdfe12e5fa537d7e8b8c6a064fe

920a3bdfe12e5f, challenge → DES → a5ef3810bf7aced4

a537d7e8b8c6a0, challenge → DES → 910af3418cd8e9af

64fe0000000000, challenge → DES → 72ad34ev019baf2d

response

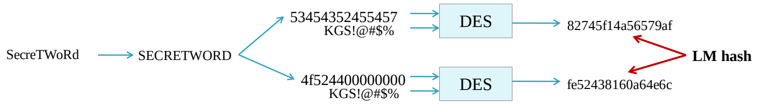
NTLM Hash, Problems

- **Problem 1:** MD4 is a very fast hash function
- **Problem 2:** No salt is used so time-memory tradeoff attacks (rainbow tables) can be used

LM Hash

Can you find problems here?

- If wanted, both NTLM and LM response are used
  - This was default before Windows Vista
- LM hash calculated as
  - Convert password to uppercase and pad to 14 bytes
  - Split into two parts of 7 byte each → two DES keys
  - Encrypt ”KGS!@#$%” with the two keys to get 16 bytes LM hash which is stored in the SAM database
- LM response calculated same way as NTLM response

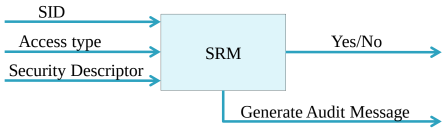


LM Hash, Problems

- **Problem 1:** DES is a fast block cipher
- **Problem 2:** No salt here either…
- **Problem 3:** Passwords up to 14 characters are never better than passwords of 7 characters
- **Problem 4:** There are no lowercase characters in the effective character set

Access Control

- Security Reference Monitor (SRM) is responsible for determining access control
- Three parameters are considered
  - Identity of subject (SID)
  - Type of access
  - Object security settings (Security Descriptor)



SID

- Security Identifier
- Unique for each user or group
- Format:

S-R-I-SA-SA-SA-N

- S: The letter S (just means that the string is a SID)
- R: revision number (1)
- I: Identifier authority (5 for user accounts)
- SA: subauthority (specifies domain or computer)
  - Can be up to 14 groups, but 3 is typical
- N: relative identifier, incremented for each new principal

Known SIDs

- Generic groups and users
- S-1-1-0 Everyone, a group that includes all users
- S-1-5-20 Network Service
- S-1-5-18 SYSTEM, local operating system
- S-1-5-SA-SA-SA-500 Administrator
- S-1-5-SA-SA-SA-501 Guest account (no password required)
- S-1-5-SA-SA-SA-512 Domain Admins (global group)

Access Token

- After successful authentication LSA builds an access token
- Processes which run as the user has a copy of the token
- When a process interacts with a securable object, token determines authorization level

User SID
SIDs of groups the user is member of
List of privileges
Default DACL, Owner, Group
Miscellaneous

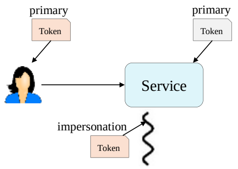
e.g., restricting SIDs

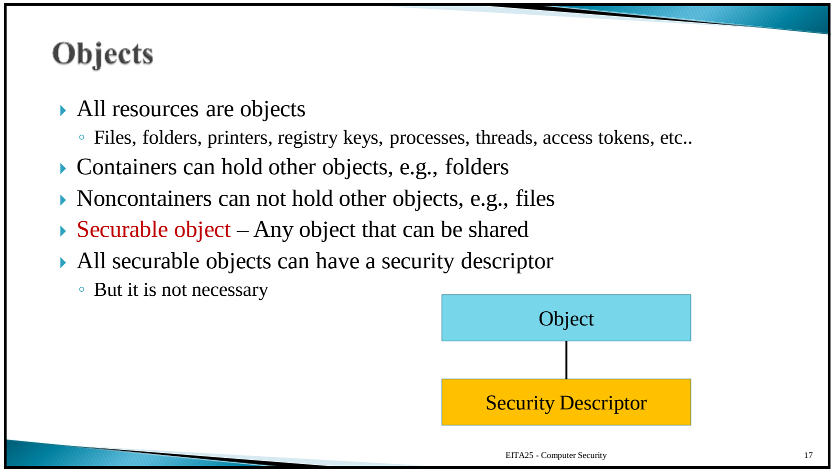
Privileges

- The right to perform system related operations
  - Shutting down
  - Change system time
  - Backup files
  - Generate audit
- Applies only to local computer. A user can have different privileges on different machines in a domain.
- Privileges can be assigned to both users and groups
- Access token is checked when user tries to perform privileged operation
- Differs from access rights
  - Access to resources and tasks, not objects
  - Stored with subject
  - Admin assigns privileges
- Stored in access token produced at logon

Two Kinds of Access Tokens

- Token is either a primary access token or an impersonation access token
- Primary access token – access token of the user account associated with the process.
  - Every process has this
- Impersonation access token – allows a thread to execute in a different security context than the process owner.
  - A thread may additionally have an impersonation access token
- **Example:** File server server runs with high privileges and can access any files
  - Threads handle concurrent user requests
  - Thread get token of user → access based on user's token
- Ability to create access token is a privilege
  - SeImporsonatePrivilege



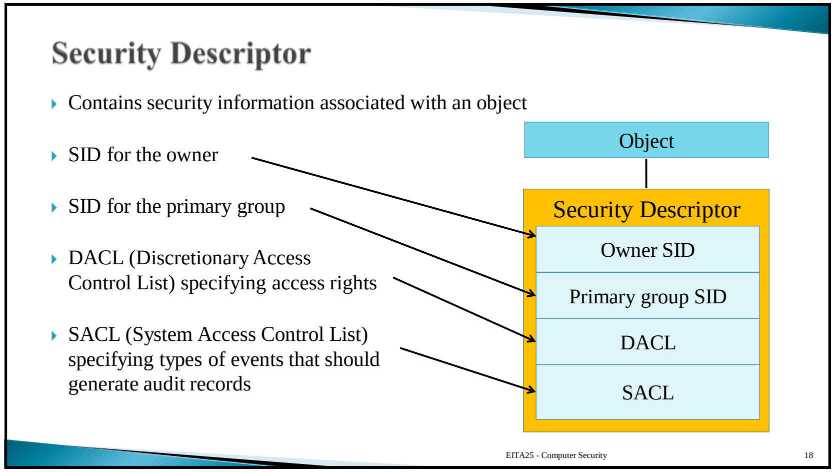


Objects

- All resources are objects
  - Files, folders, printers, registry keys, processes, threads, access tokens, etc..
- Containers can hold other objects, e.g., folders
- Noncontainers can not hold other objects, e.g., files
- Securable object – Any object that can be shared
- All securable objects can have a security descriptor
  - But it is not necessary

Object

Security Descriptor



Security Descriptor

- Contains security information associated with an object
- SID for the owner
- SID for the primary group
- DACL (Discretionary Access Control List) specifying access rights
- SACL (System Access Control List) specifying types of events that should generate audit records

Object

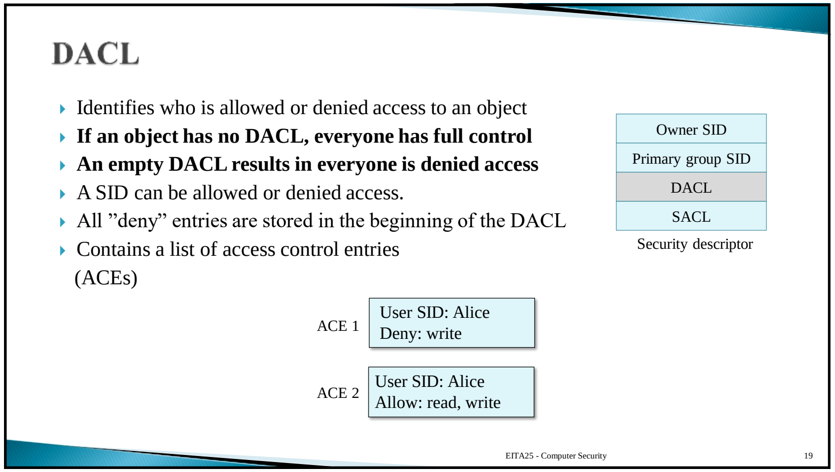
Security Descriptor

Owner SID

Primary group SID

DACL

SACL



DACL

- Identifies who is allowed or denied access to an object
- **If an object has no DACL, everyone has full control**
- **An empty DACL results in everyone is denied access**
- A SID can be allowed or denied access.
- All ”deny” entries are stored in the beginning of the DACL
- Contains a list of access control entries (ACEs)

Owner SID

Primary group SID

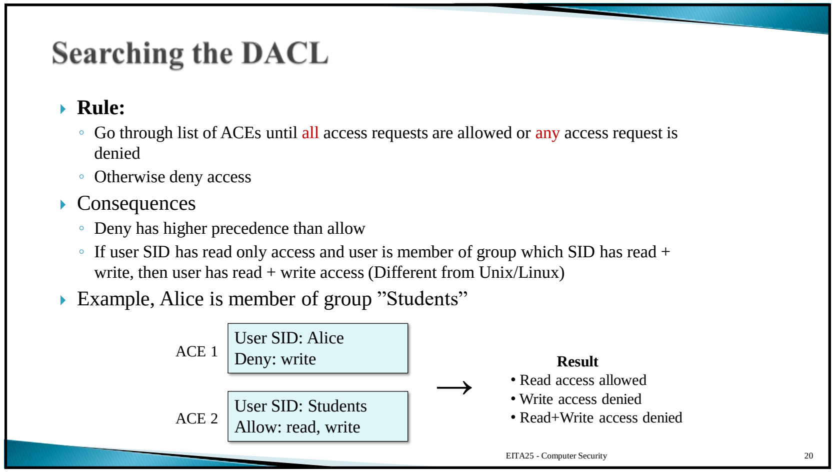
DACL

SACL

Security descriptor

ACE 1 User SID: Alice
Deny: write

ACE 2 User SID: Alice
Allow: read, write



Searching the DACL

- **Rule:**
  - Go through list of ACEs until all access requests are allowed or any access request is denied
  - Otherwise deny access
- Consequences
  - Deny has higher precedence than allow
  - If user SID has read only access and user is member of group which SID has read + write, then user has read + write access (Different from Unix/Linux)
- Example, Alice is member of group ”Students”

ACE 1 User SID: Alice
Deny: write

ACE 2 User SID: Students
Allow: read, write

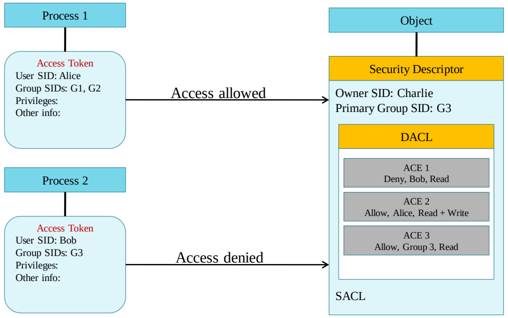
→

Result

- Read access allowed
- Write access denied
- Read+Write access denied

Example: Accessing Object

- Two processes (subjects) wants *read* access to an object



Process 1

Access Token
User SID: Alice
Group SIDs: G1, G2
Privileges:
Other info:

Access allowed

Process 2

Access Token
User SID: Bob
Group SIDs: G3
Privileges:
Other info:

Access denied

Object

Security Descriptor
Owner SID: Charlie
Primary Group SID: G3

DACL

ACE 1
Deny, Bob, Read

ACE 2
Allow, Alice, Read + Write

ACE 3
Allow, Group 3, Read

SACL

Access Rights (in the ACE)

- Since there are so many different types of objects access rights look different for different types
- **Standard access rights** apply to (almost) all objects
  - DELETE – delete the object
  - READ_CONTROL – read info in security descriptor (owner, group and DACL)
  - WRITE_DAC – write access to the DACL
  - WRITE_OWNER – write access to the field ”owner” in the security descriptor
  - SYNCHRONIZE – The right to synchronize with the object

Generic Access Rights

- Since there are many different types of objects, there are very many different types of access rights
- **Generic access rights** gives a mapping to specific access rights for a type of objects

Example – Files and directories

GENERIC_EXECUTE	FILE_EXECUTE FILE_READ_ATTRIBUTES STANDARD_RIGHTS_EXECUTE SYNCHRONIZE
GENERIC_READ	FILE_READ_ATTRIBUTES FILE_READ_DATA FILE_READ_EA STANDARD_RIGHTS_READ SYNCHRONIZE
GENERIC_WRITE	FILE_APPEND_DATA FILE_WRITE_ATTRIBUTES FILE_WRITE_DATA FILE_WRITE_EA STANDARD_RIGHTS_WRITE SYNCHRONIZE

Access Mask

- The access rights are given by a 32-bit integer

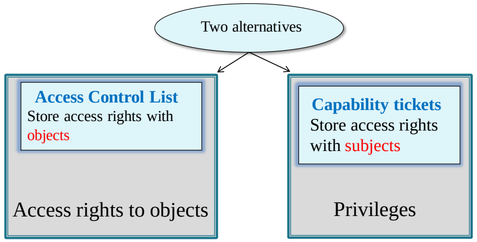
Bits	Access Right
0-15	Specific rights for the current object type
16-22	Standard rights
23	Access system security (e.g., SACL)
24-27	reserved
28	generic all
29	generic execute
30	generic write
31	generic read

Access Control, Network Shares

- Users must go through two ACL’s to access a file via a share
  - ACL on the share
  - ACL on the file itself
  - User’s effective permission through a file share is determined by masking both sets of ACL’s together.
- Example 1:
  - Client sets share permission to read only for everyone and file permission to read+write for everyone
  - Result: Users on client machine get read+write, network users get read
- Example 2:
  - Client sets share permission to full control for everyone and file permission to read for everyone
  - Result: Users on client machine get read access, network users get read access

Access Control Matrix Implementation

- Recall the two variants
  - In Windows a combination is used



However, privileges do not control access to objects.

SACL

- System Access Control List
- Controls which events should be audited
- Contains a set of ACEs that define
  - Who it applies to
  - What type of access should be logged
  - Which access result should be logged

ACE

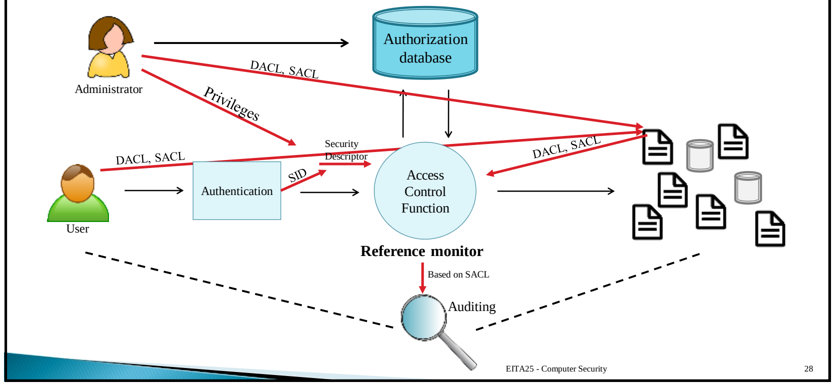
User SID: Everyone
Failure: read, write

Owner SID
Primary group SID
DACL
SACL

Security descriptor

- Also holds mandatory label (more later)

Compare with Context



The Registry

- Central database for Windows configuration data
- Just files on the harddisk
- Entries are called *keys* and *values*
- A registry *Hive* is a group of keys, subkeys, and values in the registry stored in a file
  - "Registreringsdatafil" in swedish
- Protecting the integrity of registry data is important
  - Example: The search path is set in registry, if an attacker can modify it, malicious software can be inserted/executed.
- Proprietary format: registry editor (Regedit.exe)
- Can be used by applications to store configurations

The Registry

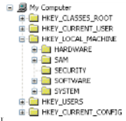
5 root keys (none is a hive)

Not symbolic links {

HKEY_CLASSES_ROOT	Merge of HKEY_LOCAL_MACHINE\SOFTWARE\CLASSES and HKEY_USERS\'SID'_Classes Contains file extension associations.
HKEY_CURRENT_USER	Symbolic link to key under HKEY_USERS that represents the user that is logged in
HKEY_LOCAL_MACHINE	Contains several hives that store information about the local computer
HKEY_USERS	Contains all active user profiles on the system.
HKEY_CURRENT_CONFIG	Symbolic link to HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Hardware Profiles\Current. Information about the hardware profile. Used when system starts up.

Some hives

Path to registry hive	Path to file hive
HKEY_LOCAL_MACHINE\SYSTEM	\WINDOWS\system32\config\system
HKEY_LOCAL_MACHINE\SAM	\WINDOWS\system32\config\sam
HKEY_LOCAL_MACHINE\SECURITY	\WINDOWS\system32\config\security
HKEY_LOCAL_MACHINE\SOFTWARE	\WINDOWS\system32\config\software



Temporary Hives

- HKEY_LOCAL_MACHINE\hardware
  - Hardware is detected when system starts
- HKEY_LOCAL_MACHINE\system\clone
  - Built during startup, saved as HKEY_LOCAL_MACHINE\SYSTEM\Select\LastKnownGood Control Set if startup is successful
  - If there is a problem to start (e.g., if an installed driver has damaged the system), then LastKnownGood configuration can be used by copying this to CurrentControlSet

Restricted Context

- Application can start process with **restricted token**
- Process can start process or thread with restricted token
  - Can be either primary token or impersonation token
- Example 1: Untrusted webpages can be displayed with restrictions
- Example 2: Email attachments can be opened with restrictions
- Restrict by (one or more of):
  1. Remove privileges
  2. Set deny-only attribute to SIDs
  3. Specify restricting SID

How To Restrict a Token

1. Let group SIDs be used for deny only
2. Add restricted SID

→ Two access checks are done

Both must allow access, otherwise access is denied

Restricted Token	
User SID	Alice
Group SIDs	Admin (deny only)
	Users
Restricted SIDs	SID_Restr
Privileges	none
…	

Example: Process with restricted token require read access

a) Principal SID: Alice
Allow: read, write

Principal SID: SID_Restr
Allow: read

Access granted

b) Principal SID: Admin
Allow: read

Principal SID: SID_Restr
Allow: read

Access denied

c) Principal SID: Alice
Allow: read, write

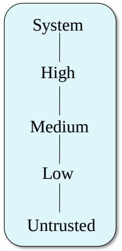
Access denied

User Account Control (UAC)

- Introduced in Windows Vista
- Administrators get two access tokens when logging in
  - One administrator token
  - One standard user token
- Standard user token used unless administrator privileges are needed
  - User has to actively acknowledge use of administrator token
- Windows 7+ uses UAC, but not all programs ask for explicit permission

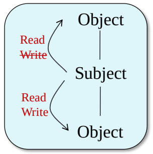
Mandatory Access Control

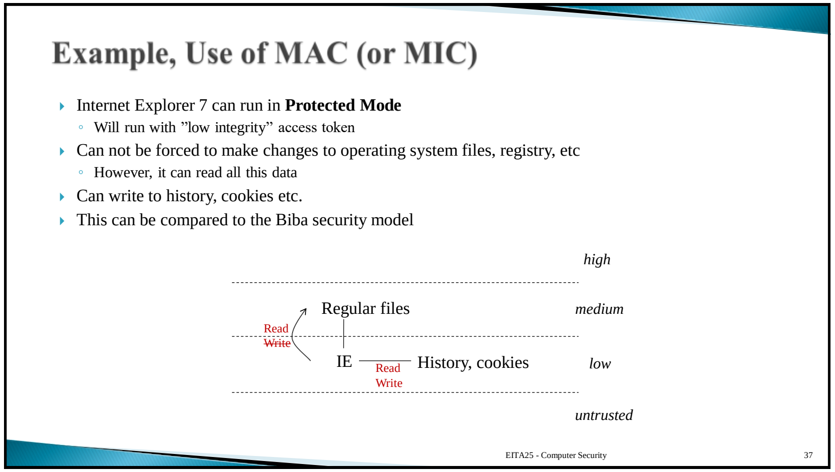
- Windows Vista and later include mandatory access control (MAC)
  - Called Integrity Control
- **Access tokens** have an integrity level
  - Untrusted (Processes started by group Anonymous)
  - Low integrity (e.g., IE in protected mode)
  - Medium integrity (Used by normal applications when UAC is enabled)
  - High integrity (Admin applications started through UAC, normal applications if UAC is disabled)
  - System integrity (Used by some system processes)



Mandatory Access Control

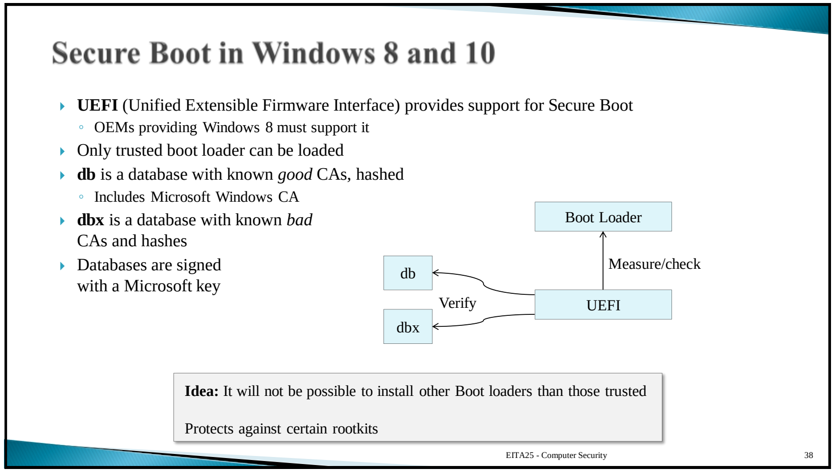
- Each object can also have an integrity level stored in the **Security Decriptor**'s SACL
- Default for newly created objects:
  - If access token is lower than medium, integrity level of object is same as in access token
  - If access token is medium or higher, integrity level of object is medium
- Subject has label S, object has label O
- **Policy defined by (total) ordering:**
  - Write access granted if O ≤ S
- Subjects integrity level must dominate object's integrity level in write operations
  - Checked before DACL





Example, Use of MAC (or MIC)

- Internet Explorer 7 can run in **Protected Mode**
  - Will run with ”low integrity” access token
- Can not be forced to make changes to operating system files, registry, etc
  - However, it can read all this data
- Can write to history, cookies etc.
- This can be compared to the Biba security model



Secure Boot in Windows 8 and 10

- **UEFI** (Unified Extensible Firmware Interface) provides support for Secure Boot
  - OEMs providing Windows 8 must support it
- Only trusted boot loader can be loaded
- **db** is a database with known *good* CAs, hashed
  - Includes Microsoft Windows CA
- **dbx** is a database with known *bad* CAs and hashes
- Databases are signed with a Microsoft key

Idea: It will not be possible to install other Boot loaders than those trusted

Protects against certain rootkits