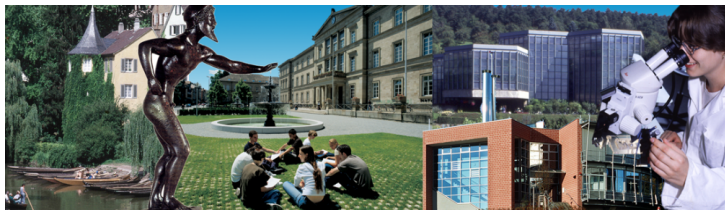


EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN



Introduction to Computer Security

Windows Security

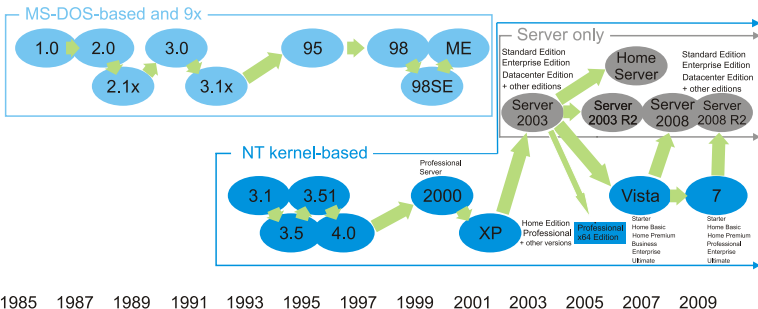
Pavel Laskov

Wilhelm Schickard Institute for Computer Science

<https://t.me/learningnets>



Microsoft Windows Family Tree



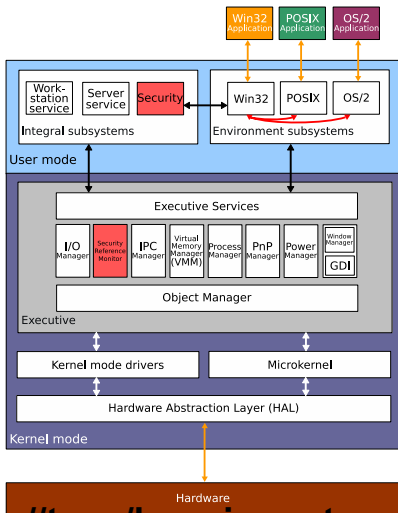
Key security milestones:

- NT 3.51 (1993): network drivers and TCP/IP
- Windows 2000: Active Directory, Kerberos, security architecture.
- Server 2003: security policies, LAN and wireless security
- Vista (2007): no “admin-by-default”, firewall, DEP, ASLR
- 64-bit versions (Vista+): mandatory kernel code signing

<https://t.me/learningnets>



Architecture of modern Windows OS



<https://t.me/learningnets>



Security components of Windows OS

- Kernel mode:
 - Security Reference Monitor: ACL verification
- User mode:
 - Log-on process (winlogon): user logon
 - Local Security Authority (LSA): password verification and change, access tokens, audit logs (MS04-11 buffer overflow: Sasser worm!)
 - Security Accounts Manager (SAM): accounts database, password encryption
 - User Account Control (UAC, Vista): enforcement of limited user privileges

<https://t.me/learningnets>



Windows registry

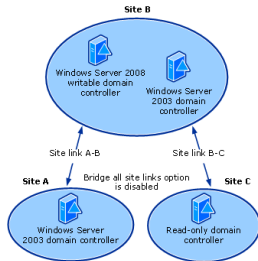
- A hierarchical database containing critical system information
- Key-value pairs, subkeys, 11 values types
- A registry **hive** is a group of keys, subkeys, and values
- Security-related registry hives:
 - HKEY_LOCAL_MACHINE\SAM: SAM database
 - HKEY_LOCAL_MACHINE\Security: security logs, etc
 - HKEY_LOCAL_MACHINE\Software: paths to programs!
- Security risks:
 - manipulated registry entries
 - missing security-related registry keys

<https://t.me/learningnets>



Windows domains

- A **domain** is a collection of machines sharing user accounts and security policies.
- Domain authentication is carried out by a **domain controller (DC)**.
- To avoid a single point of failure, a DC may be replicated



<https://t.me/learningnets>



Active directory

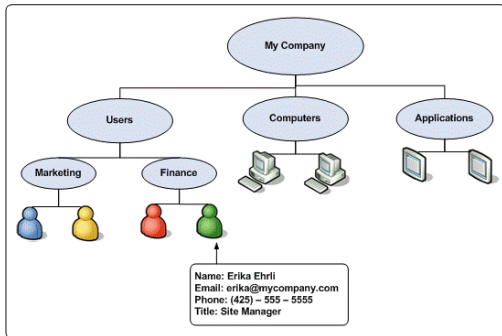
Active directory introduced in Windows 2000 is an LDAP-like directory service for organization of system resources:

- Users and groups
- Security credentials and certificates
- System resources (desktops, servers, printers)
- Security policies
- DNS service
- Trust management

<https://t.me/learningnets>



Active Directory structure



<https://t.me/learningnets>



Access control in Windows

- Access control is applied to **objects**: files, registry keys and hives, Active Directory objects.
- More than just access control on files!
- Various means exist for expressing security policies:
 - groups
 - roles
 - ownership and inheritance rules
 - complex access rights

<https://t.me/learningnets>



- Principals are active entities in **security policies**
- Principals can be
 - local users
 - aliases
 - domain users
 - groups
 - machines
- Principals have a human readable **user name** and a unique **security identifier (SID)**
- **Local principals** are created by a LSA, e.g.
principal = MACHINE\principal
- **Domain principals** are administered by DC, e.g.
principal@domain = DOMAIN\principal

<https://t.me/learningnets>



Security identifiers

- A **security identifier (SID)** is a **unique**, machine generated code of varying length used to identify principals.
- Format: S-1-IA-SA-SA-SA-N, where
 - IA (identifier authority): characterizes an issuer, e.g. World Authority (1) or Local Authority (2)
 - SA (subauthority): identifies a specific SID issuer, e.g. a domain controller
 - N: relative identifier, unique for each authority
- Examples:
 - Everyone (World): S-1-1-0
 - System: S-1-5-18
 - Administrator: S-1-5-21-<domain>-500

<https://t.me/learningnets>



Principals used for access control

- **SID**: an individual principal
- **Group**: a collection of principals managed by DC; groups have their own SIDs and can be nested
- **Alias**: a local group managed by LSA; cannot be nested
- **Aliases implement logical roles**: an application may define an alias to which SIDs are assigned at run-time

<https://t.me/learningnets>



Subjects

- Subjects are active entities in OS primitives.
- Windows subjects are processes and threads.
- Security credentials for a subject are stored in a token.
- Tokens provide a principal/subject mapping and may contain additional security attributes.
- Tokens are inherited (possibly with restrictions) during creation of new processes.

<https://t.me/learningnets>



Token contents

- Identity and authorisation contents
 - user SID, group SIDs, alias SIDs
 - privileges
- Defaults for new securable objects
 - owner SID, group SID, DACL
- Miscellaneous attributes
 - logon SID

<https://t.me/learningnets>



- A set of fixed **privileges** is defined by numeric constants in `Winnt.h`
- Privileges control access to system resources.
- Example privileges:
 - load or unload a device driver
 - lock a page in a physical memory
 - create a computer account
 - shut down a system
 - modify a system time
- Privileges are **not** access rights!

<https://t.me/learningnets>



- **Objects** represent various passive OS entities
- Example Windows objects:
 - files or directories
 - pipes
 - processes and threads
 - file mappings
 - access tokens
 - window-management objects
 - registry keys
 - printers
 - network shares
 - synchronization objects
 - job objects
 - Active Directory objects
- Security of built-in objects is managed by OS
- Security of private objects must be managed by applications
- **Securable objects** are equipped with a **security descriptor**

<https://t.me/learningnets>



Security descriptor

Owner SID
Primary Group SID
DACL
SACL

- Owner: a principal who owns an object
- Primary group: for POSIX compatibility
- DACL: specifies who is granted and who is denied access
- SACL: specifies a security audit policy

<https://t.me/learningnets>



Access rights: an overview

- Describe what one can do to an object
- Encoded as a 32-bit mask
- **Standard access rights (bits 16–23)** are common to most object types
 - DELETE
 - READ_CONTROL: read object's security descriptor
 - SYNCHRONIZE: use object for synchronization (not all objects)
 - WRITE_DAC: change object's DACL
 - WRITE_OWNER: change object's owner
- **Object-specific rights (bits 0–15)** are tailored to each class of objects
- **Extended rights** can be specified for Active Directory entries.

<https://t.me/learningnets>



Generic access rights

- The highest 4 bits (28–31) represent **generic access rights**:
 - `GENERIC_READ`
 - `GENERIC_WRITE`
 - `GENERIC_EXECUTE`
 - `GENERIC_ALL`
- Each class of objects maps its generic rights to object-specific rights.
- Generic rights are used to simplify design: they provide an intermediate description level for access rights.

<https://t.me/learningnets>



ACLs in Windows

- DACL in a security descriptor is a list of **Access Control Entries (ACE)**
- ACE format:
 - ACE type: positive or negative permissions
 - Principal SID
 - Access rights mask
 - Inheritance flags
- ACEs are processed **sequentially** until either some entry denies all requested access rights or a set of ACEs grants all requested access rights

<https://t.me/learningnets>



ACE matching algorithm

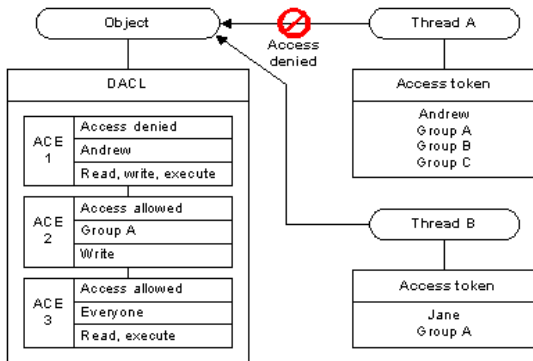
For any objects that **do not have** DACL, access is always granted. For all other objects, the subject's token is compared sequentially with each ACE as follows:

- ACE does not contain a matching SID: skip and continue.
- SID matches and contains a negative permission: deny access and stop.
- SID matches and contains a positive permission:
 - if accumulated access rights match access mask, grant access and stop.
 - otherwise add ACE to the accumulated access rights and continue.

<https://t.me/learningnets>



ACL processing example

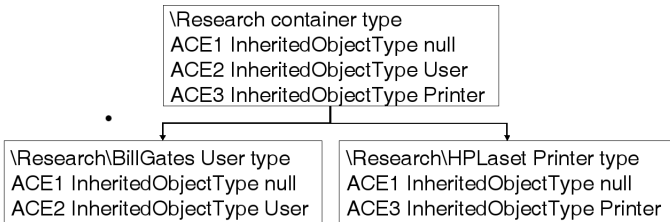


<https://t.me/learningnets>



ACL inheritance

- When a new object is created, its ACL is inherited from that of the enclosing container, e.g. a directory.
- Only ACEs with a matching object type are inherited



<https://t.me/learningnets>



Summary

- Windows systems contain complex security mechanisms for user account and object management as well as access control.
- DACLs enable fine-grained access control of heterogeneous entities in Windows itself as well as applications.
- The complexity of Windows security mechanisms is also its enemy: misconfiguration as well as implementation bugs may lead to severe security incidents.

<https://t.me/learningnets>