# Unit 7: Security

## 7.2. Windows Security Components and Concepts

# Roadmap for Section 7.2.

- Windows Security Features
- Components of the Security System
- Windows Logon
- Kerberos Protocol Principles / Active Directory

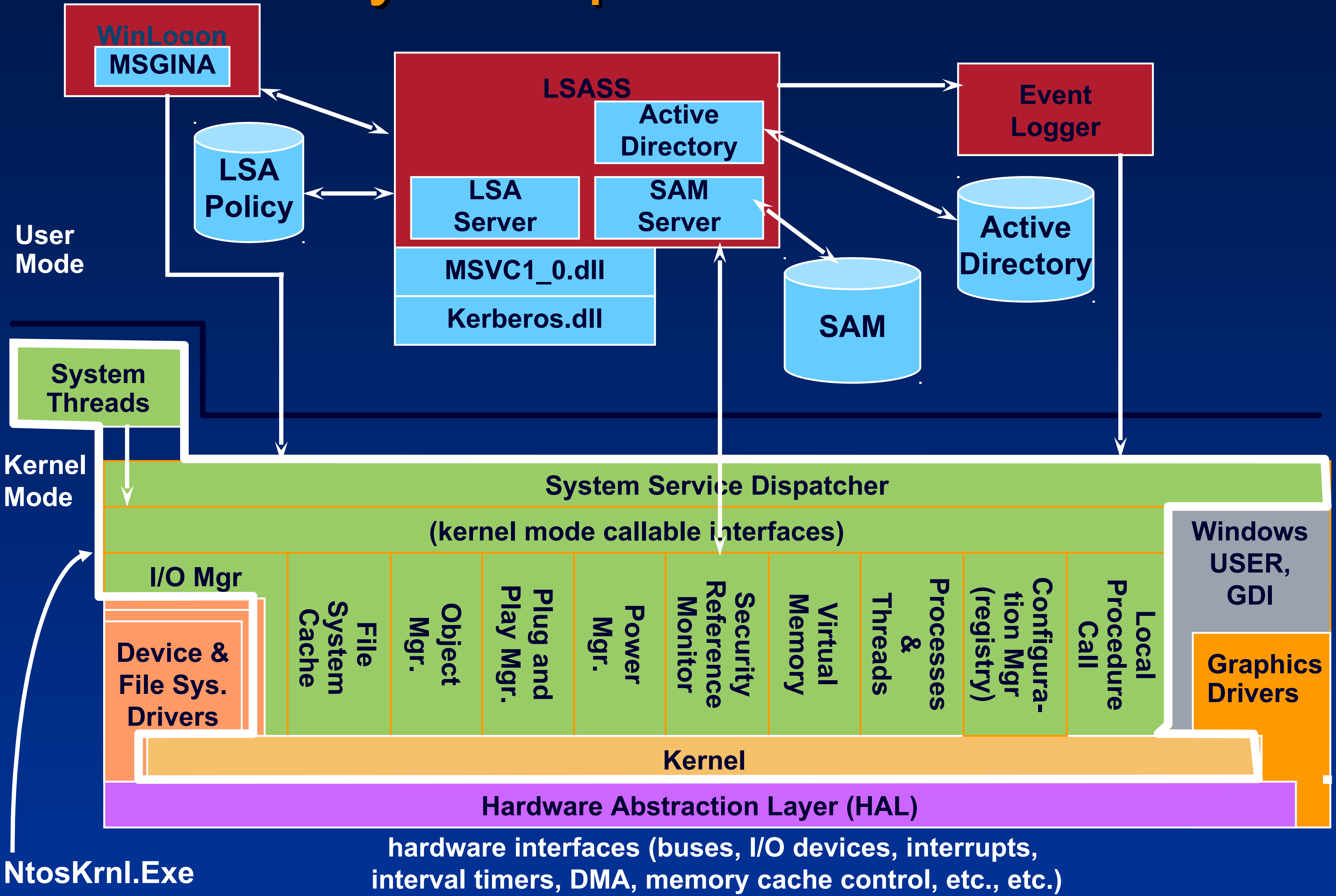
# Windows Security Mechanisms

- Permissions can be applied to all shareable resources
  - Including the NTFS file system
  - …but not the FAT file system
- Encrypted File System protects data while OS is offline
  - Un-authorized physical access
- Native support for Kerberos authentication
- Public Key infrastructure to pass digital certificates
- IP Security to protect sensitive data traveling across the wire
- Crypto-APIs built into Windows
  - Hashing and encryption

# The three hearts of Windows Security

- Local Security Authority (LSA) – as a local user-mode process
  - Heart of user authentication on local machine
- LSA - on domain controller
  - Heart of user authentication on networked machines
- Security Reference Monitor – as a component from executive (i.e. kernel-mode component)
  - Heart of object access protection

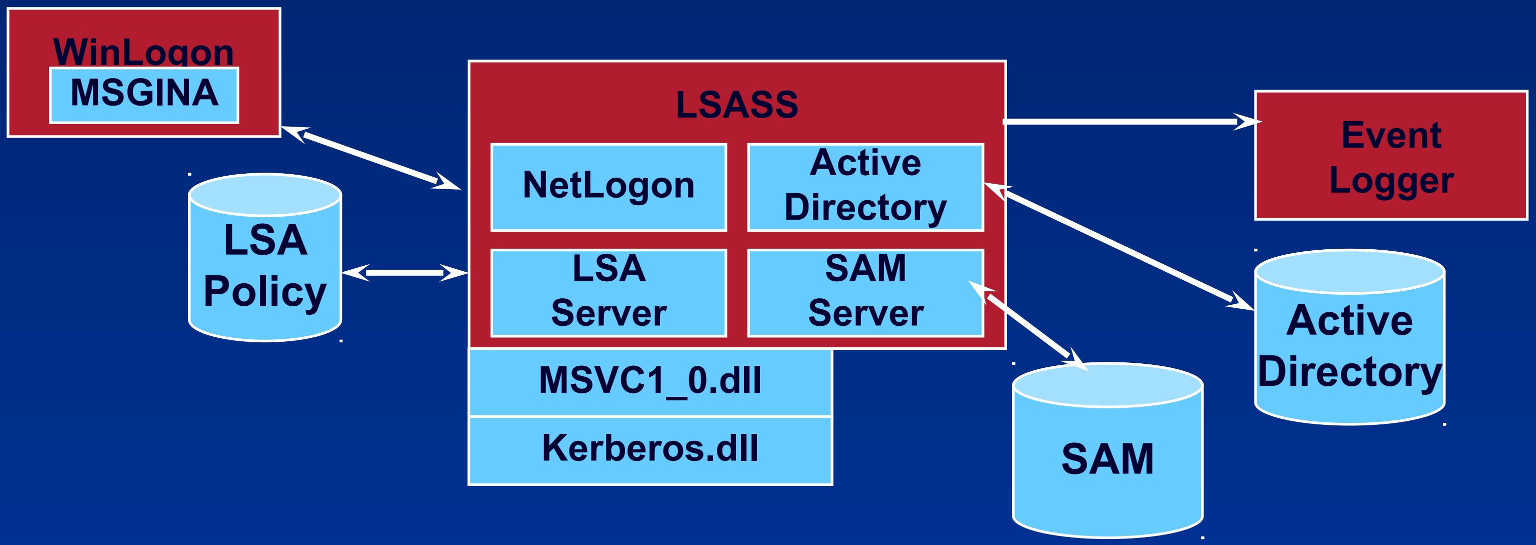
# Security Components



hardware interfaces (buses, I/O devices, interrupts, interval timers, DMA, memory cache control, etc., etc.)

# 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



# LSASS Components

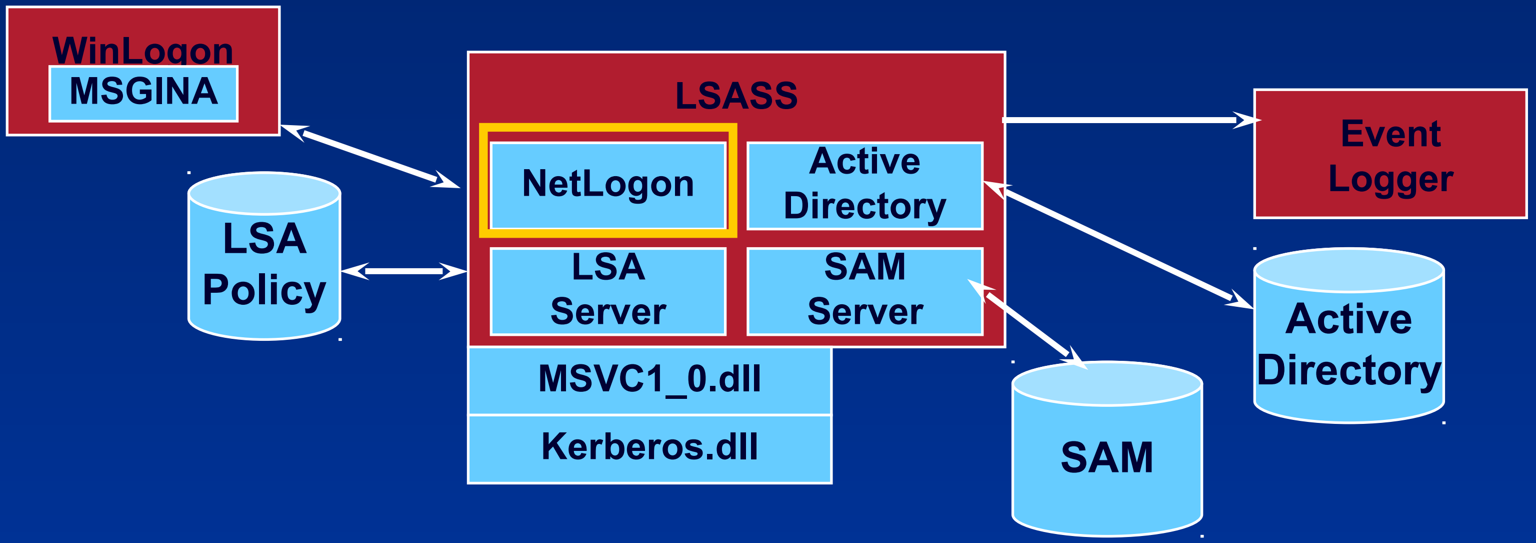
- SAM Service
  - A set of subroutines (\Windows\System32\Samsrv.dll ) responsible for managing the database that contains the usernames and groups defined on the local machine
  - SAM database: A database that contains the defined local users and groups, along with their passwords and other attributes. This database is stored in the registry under HKLM\SAM.
  - Password crackers attack the local user account password hashes stored in the SAM
- Lab: look at SAM service
  - Open Lsass.exe process properties – click on services tab
  - Click Find DLL – search for Samsrv.dll

# LSASS Components

- Active Directory
  - A directory service that contains a 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
  - The Active Directory server, implemented as a service, \Windows\System32\Ntdsa.dll, that runs in the Lsass process
- Authentication packages
  - DLLs that run in the context of the Lsass process and that implement Windows authentication policy:
    - LanMan: \Windows\System32\Msvc1_0.dll
    - Kerberos: \Windows\System32\Kerberos.dll
    - Negotiate: uses LanMan or Kerberos, depending on which is most appropriate

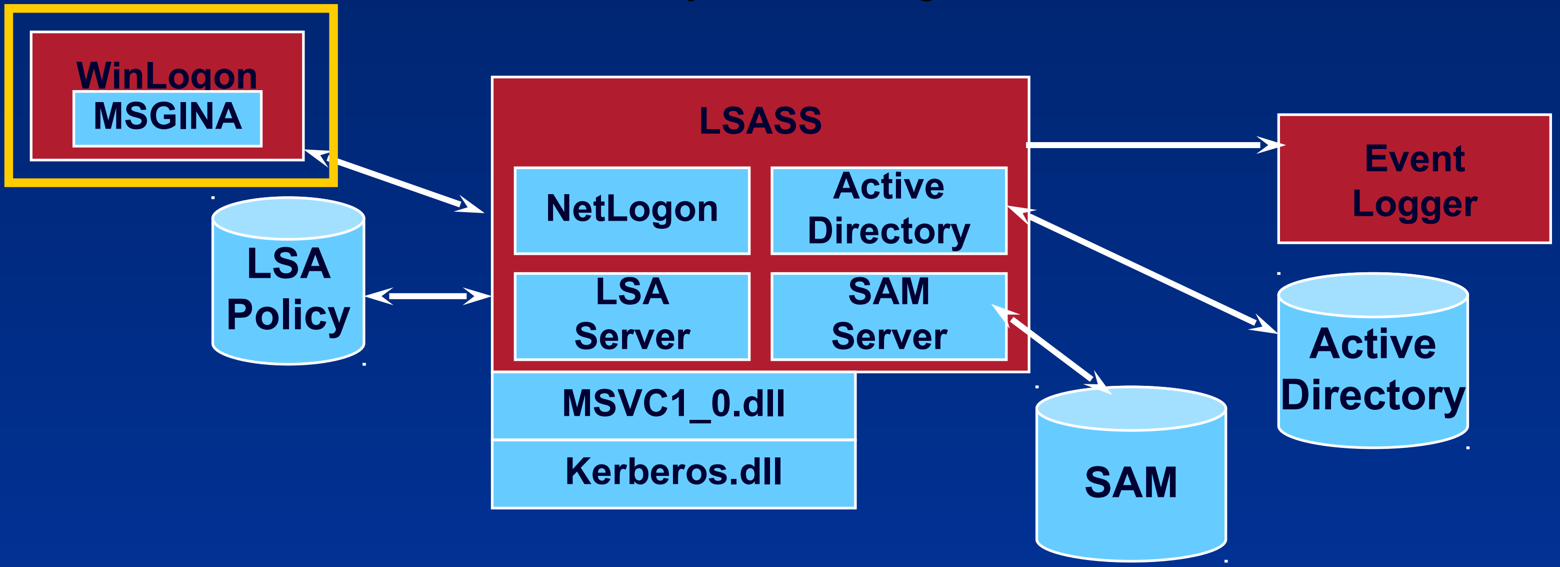
# LSASS Components

- 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 SAS (i.e. Ctrl+Alt+Del sequence) 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

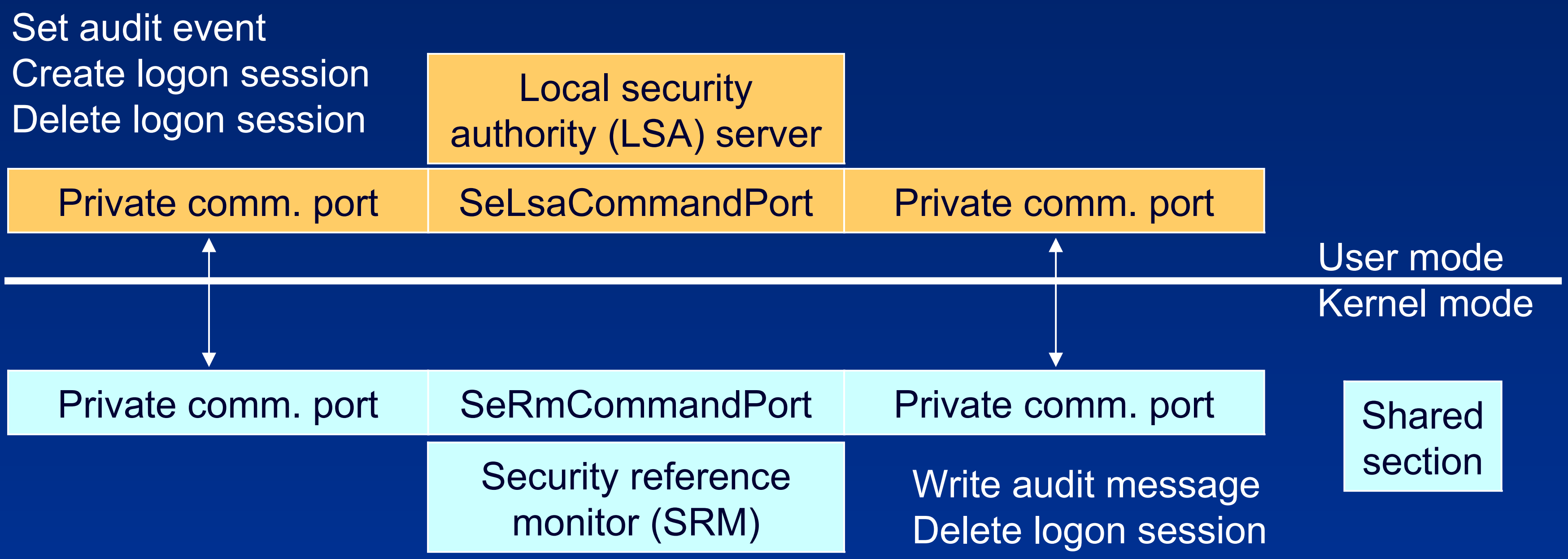


# Security Reference Monitor

- Performs object access checks, manipulates privileges, and generates audit messages
- Group of functions in Ntoskrnl.exe
    - Some documented in DDK
    - Exposed to user mode by Windows API calls
- Lab: Open Ntoskrnl.exe with Dependency Walker and view functions starting with “Se”

# Communication between SRM and LSA

- Communication via local procedure call (LPC)
  - SeLsaCommandPort/SeRmCommand port for initialization
  - Usage of private ports/shared memory when initialization is completed



# What Makes Logon Secure?

- Before anyone logs on, the visible desktop is Winlogon’s
- Winlogon registers CTRL+ALT+DEL, the Secure Attention Sequence (SAS), as a standard hotkey sequence
- SAS takes you to the Winlogon desktop
- No application can deregister it because only the thread that registers a hotkey can deregister it
- When Windows’ keyboard input processing code sees SAS it disables keyboard hooks so that no one can intercept it

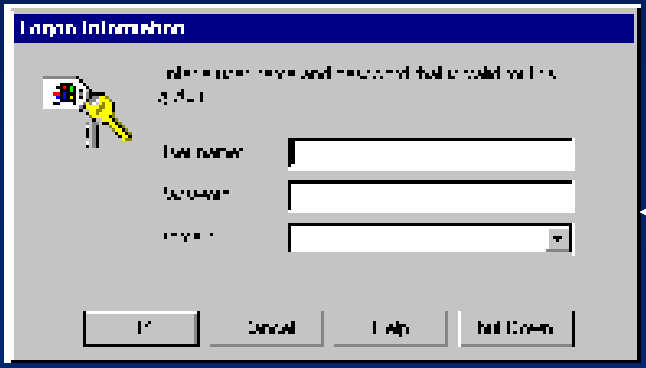
# Logon

- After getting security identification (account name, password), the GINA sends it to the Local Security Authority SubSystem (LSASS)
- LSASS calls an authentication package to verify the logon
  - If the logon is local or to a legacy domain, MSV1_0 is the authenticator. User name and password are encrypted and compared against the Security Accounts Manager (SAM) database
  - If the logon is to a AD domain the authenticator is Kerberos, which communicates with the AD service on a domain controller
- If there is a match, the SIDs of the corresponding user account and its groups are retrieved
- Finally, LSASS retrieves account privileges from the Security database or from AD

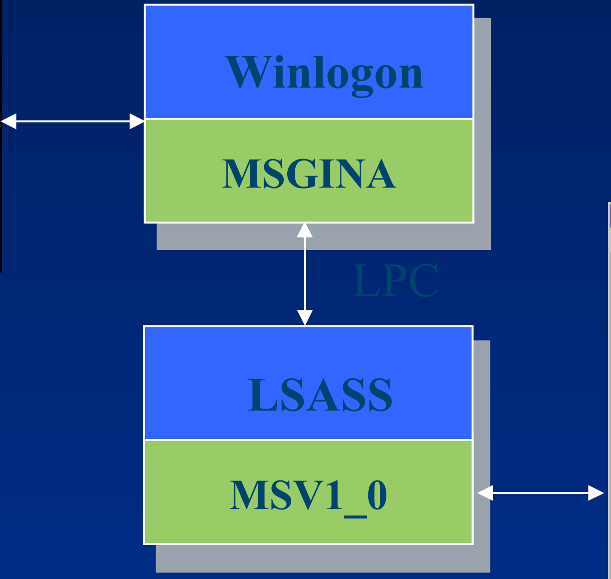
# Logon

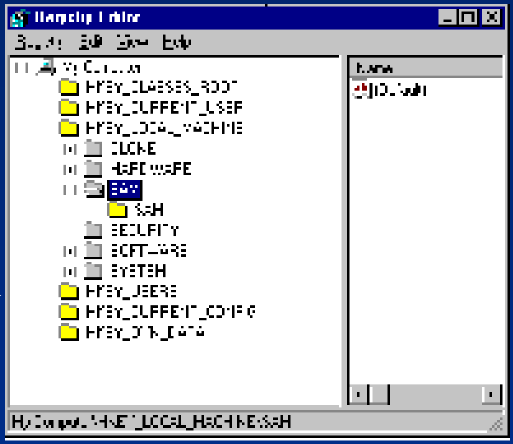
- LSASS creates a token for your logon session and Winlogon attaches it to the first process of your session
  - Tokens are created with the NtCreateToken API
  - Every process gets a copy of its parent’s token
- SIDs and privileges cannot be added to a token
- A logon session is active as long as there is at least one token associated with the session
- Lab
  - Run “LogonSessions –p” (from Sysinternals) to view the active logon sessions on your system

# Local Logon



Enter a user name and password that is valid for this system.



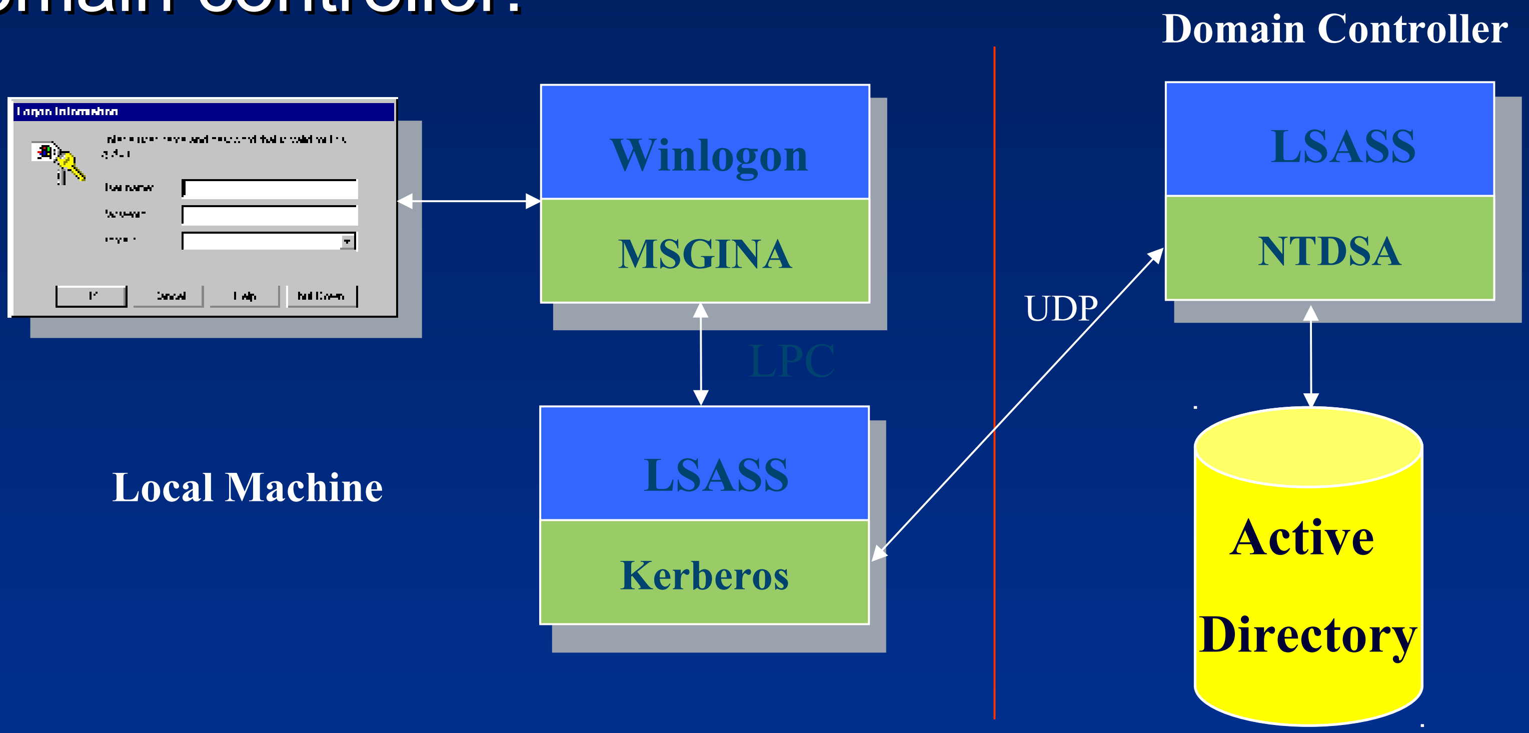


My Computer\HKEY_LOCAL_MACHINE\SAM

SAMSRV

# Remote Logon - Active Directory

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



Domain Controller

Winlogon

MSGINA

LPC

LSASS

Kerberos

Local Machine

UDP

LSASS

NTDSA

Active Directory

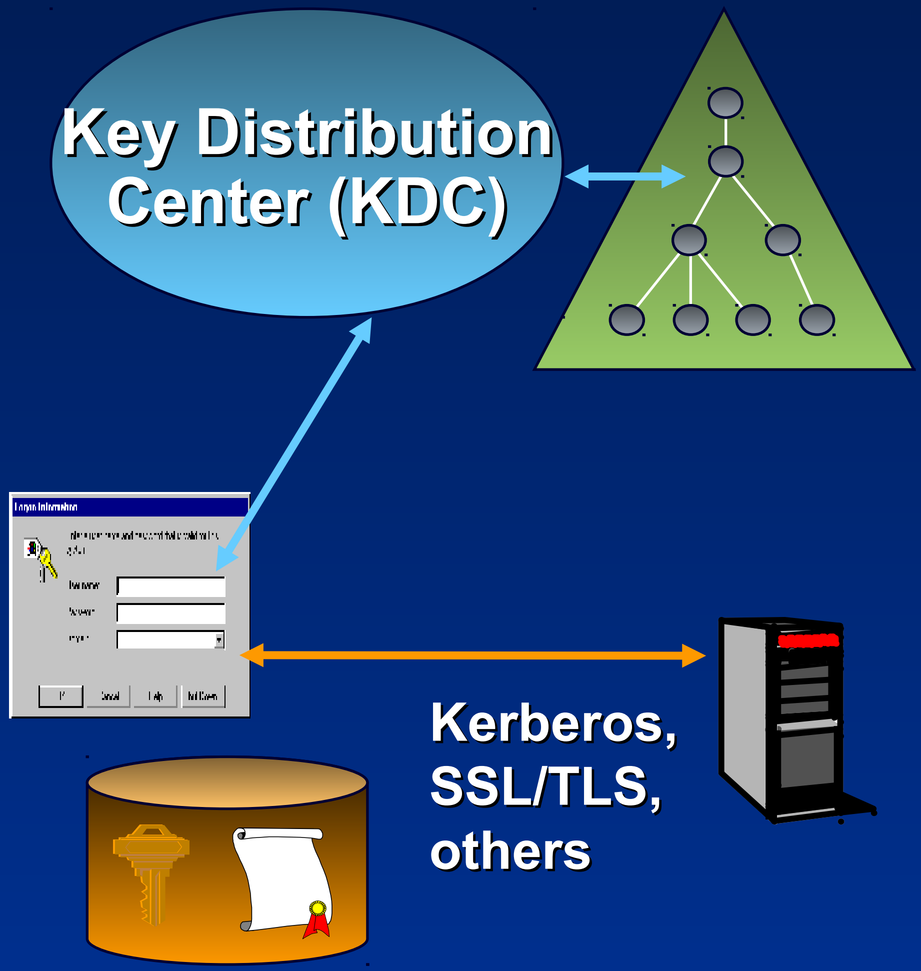
# Kerberos Authentication

- Single account store in Active Directory

- Integrated Kerberos v5 logon

- Protected store for public key credentials

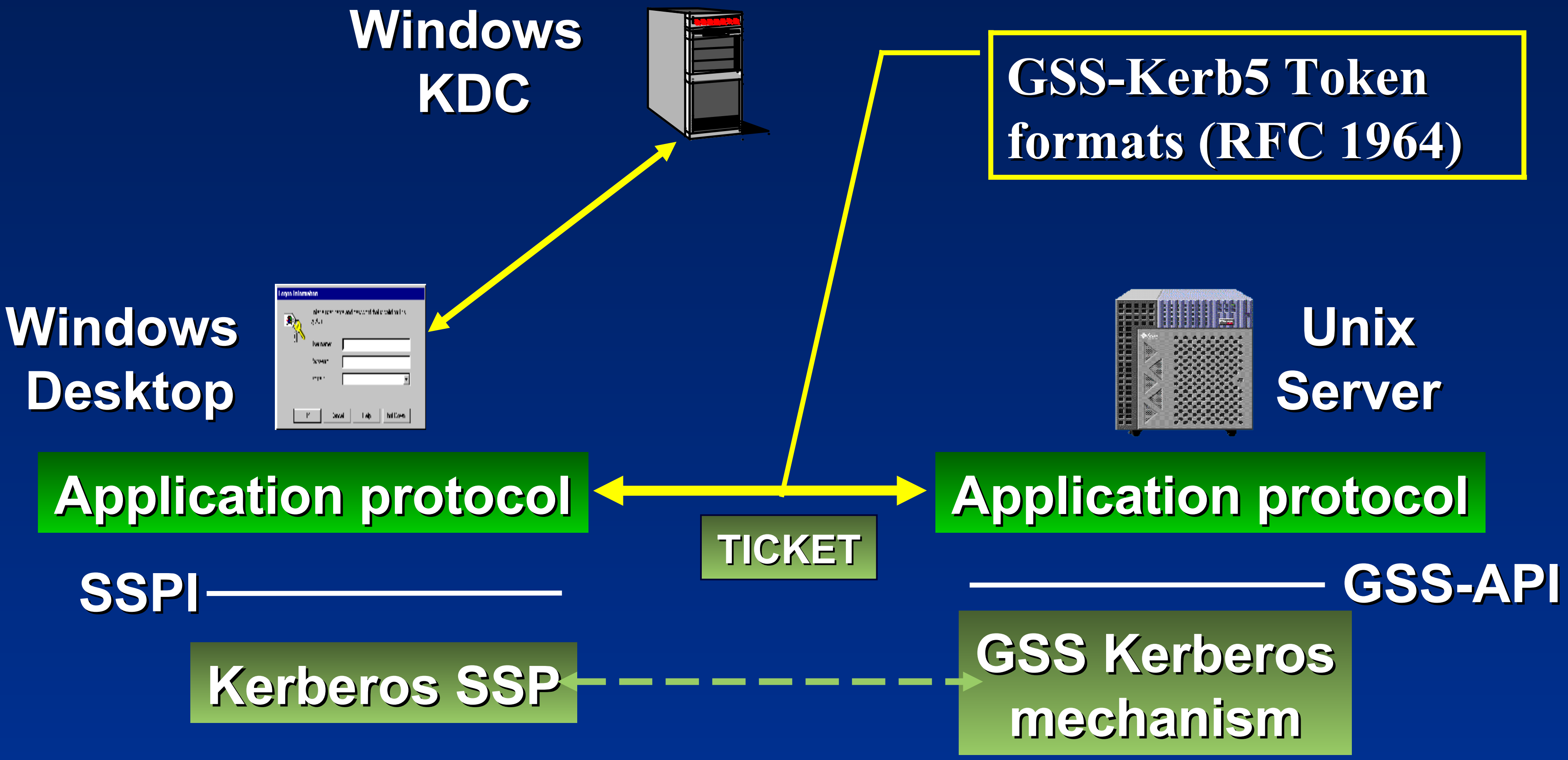
- Industry standard network security protocols



( SSL - Secure Socket Layer, TLS - Transport Layer Security )

# Cross-platform Strategy

- Common Kerberos domain



Windows
KDC

GSS-Kerb5 Token
formats (RFC 1964)

Windows
Desktop

Unix
Server

Application protocol

TICKET

Application protocol

SSPI

GSS-API

Kerberos SSP

GSS Kerberos
mechanism

( SSPI - Security Service Provider Interface, GSS - Global Security Service )

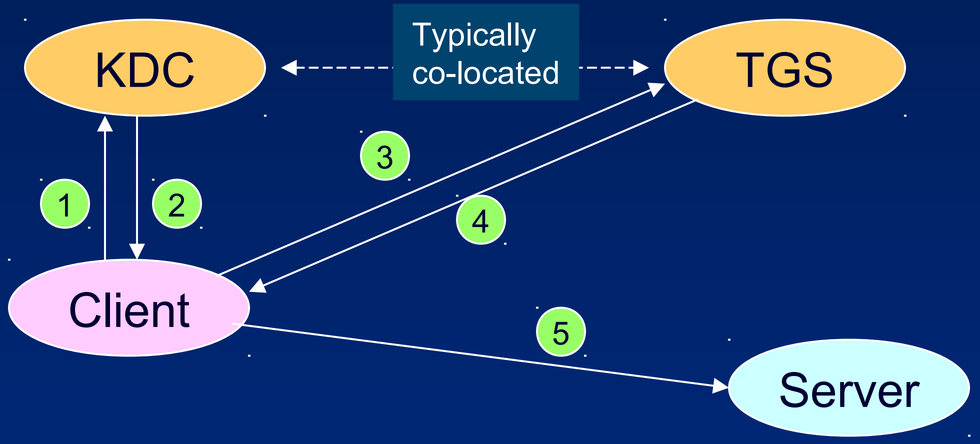
# Kerberos Authentication Service

- Developed as part of MIT project Athena
- Kerberos implements an authentication procedure which verifies identity of communication partners
  - DES algorithm, symmetric key encryption
  - Authentication server (Kerberos Server)
  - TGS (Ticket Granting Service)
  - Client proves his identity by presenting an encrypted, service-specific ticket ($T_{c,s}$) when issuing a request
- Kerberos server and Ticket Granting Service (TGS) are assumed to be secure (trusted hosts)

# Kerberos principles

- Kerberos requires three main steps:
  1. Client identifies himself against Kerberos Server (Active Directory), it receives a master ticket – the Ticket Granting Ticket (TGT)
  2. Client requests service-specific tickets and proves his identity with the TGT
  3. Client uses service-specific ticket to contact server
- Authentication is transparent from user‘s point of view
  - Windows login program acquires TGT
  - (Client) Applications transparently acquire service-specific tickets
  - TGS-issued tickets and TGT have a default lifetime of eight hours

# Kerberos principles (contd.)



1. Client → KDC: c, tgs, n
2. KDC → Client: $\{K_{c,tgs},n\}K_c$, $\{T_{c,tgs}\}K_{tgs}$
3. Client → TGS: $\{A_c\}K_{c,tgs}$ , $\{T_{c,tgs}\}K_{tgs}$, s, n
4. TGS → Client: $\{K_{c,s}, n\}K_{c,tgs}$ , $\{T_{c,s}\}K_s$
5. Client → Server: $\{A_c\}K_{c,s}$ , $\{T_{c,s}\}K_s$

| | |
|---|---|
| $K_{c:}$ | client‘s secret key |
| $K_{c,tgs:}$ | key for communication between client and TGS |
| $\{T_{c,tgs}\}K_{tgs:}$ | encrypted ticket for TGS |
| $K_{c,s:}$ | key for client/service communication |
| $\{T_{c,s}\}K_{s:}$ | encrypted ticket for service |
| $A_{c:}$ | authentication info |

# Tickets and Authentication info

- Kerberos tickets contain the following data:
  - User name
  - Address of workstation
  - Time stamp
  - Lifetime of the ticket
  - Address of the host running the requested service
  - Session key for client/server communication
- Tickets are encrypted with the server‘s private key ($K_s$)
- Authentication info ($A_c$) contains the following data:
  - User name
  - Address of workstation
  - Time stamp
- Authentication info is encrypted with the session key $K_{c,s}$

# Kerberos Version 5 - Windows

- Multiple supported encryption algorithms through Crypto-API foundation
- Keys carry info about encryption algorithm used
  - Can be re-used for different encryption algorithms
- Network addresses may have arbitrary formats
  - Server may specify all supported protocols/addresses in ticket
- Network data format and encryption are standardized
  - ASN.1 format (ISO 8824), no special format for multi-byte data
  - Encryption based on (ISO 8825)
- Tickets contain plaintext section
  - Server may support multiple personalities, actual role is chosen on plaintext info
- Tickets carry starting time and expiration time

# Ticket Characteristics

- KDC returns special tickets on initial ticket exchange
  - Password can only changed with those special tickets
- Renewable tickets may carry two expiration dates
  - Only valid after first but before second date
- Tickets may be postdated
  - Interesting for batch processing
- Authorization data field
  - KDC copies authorization info from TGT into every newly generated ticket
  - Windows Kerberos supports public/private key for initial authentication (to obtain TGT via user-supplied private key)

# Further Reading

- Pavel Yosifovich, Alex Ionescu, et al., “Windows Internals”, 7th Edition, Microsoft Press, 2017.
    - Chapter 6 – Security (from pp. 837)
        - Security system components (from pp. 840)
        - Logon (from pp. 983)
- John T.Kohl, B.Clifford Neumann, Theodore Y.Ts’o, *The Evolution of the Kerberos Authentication Service*, Proceedings of Spring 1991 EurOpen Conference, Tromsø, Norway.
- The Open Software Foundation, *Introduction to DCE*, Prentice-Hall, 1992.
- The Open Software Foundation, *DCE User’sGuide and Reference*, Prentice-Hall, 1992.