

Windows Security

Module 1 - Security Mechanisms



External resources disclaimer

This material includes links to external publicly available articles, projects, and research papers which are provided to you as a convenience and for informational purposes only.

Microsoft bears no responsibility for the accuracy, legality, content or any other aspect of the external site. Use of external hyperlinks does not constitute an endorsement by Microsoft of the linked content.

The external content referenced in this document belongs exclusively to their respective author(s). Inclusion in this presentation does not grant you with any right on the external content. You must comply with the original source's applicable policies.

How to use this document

Why this document?

This document is provided as a companion of the video lessons. Additional information is included here which would not fit the video format or would exaggeratedly lengthen the videos. As you are watching the videos, the instructor will point you to additional content in this document.

Structure

The structure of this slide deck follows the structure of the lessons. One slide deck is provided for each module. The slide deck has the same structure (naming of chapters and sections) than the video so that you can quickly jump to the slides associated with the lesson you're currently watching.

Section

1

Authentication services



Agenda

- _____ Fundamental Windows identity structures
- _____ SAM and LSASS services
- _____ SSPI
- _____ User session lifecycle
- _____ Credential Manager
- _____ UAC
- _____ gMSA

Chapter

1.1.1

Fundamental identity structures



Fundamental structures



SID – Security IDentifier



Access Token

SID – Security IDentifier

What

A security identifier (SID) is a unique value of variable length used to identify a trustee.

Who

Each account has a unique SID issued by an authority, such as a Windows domain controller, the local machine, the service manager, or any third-party package responsible to authenticate a trustee.

Where

Users' SID are stored in the SAM – *Security Account Manager* which is the Windows' identity database both for isolated machines and Active Directory domains.

Why

The system uses the SID in the access token to identify the user in all subsequent interactions with Windows security.

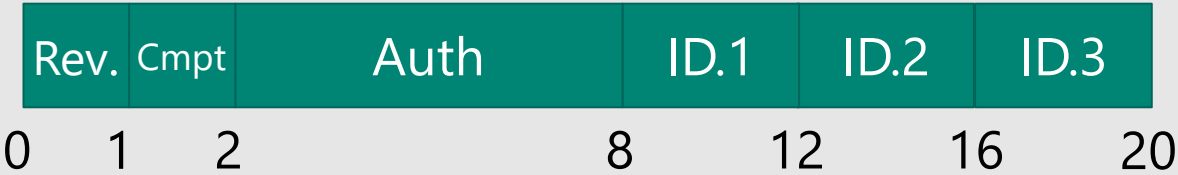
Security Identifiers structure

Logical form



Last DWORD also called RID

Binary form



Text representation

S-<Rev.>-<Auth>-<ID.1>-...-<ID.n>

Maximum 15 SubAuthority DWORDS

SID definition

C

```
typedef struct _SID_IDENTIFIER_AUTHORITY {  
    UCHAR Value[6];  
} SID_IDENTIFIER_AUTHORITY, *PSID_IDENTIFIER_AUTHORITY;
```

```
typedef struct _SID {  
    UCHAR Revision;  
    UCHAR SubAuthorityCount;  
    SID_IDENTIFIER_AUTHORITY IdentifierAuthority;  
    ULONG SubAuthority[ANYSIZE_ARRAY];  
} SID, *PISID;
```

SID Examples



Windows Service

S-1-5-80-859482183-879914841-863379149-1145462774-2388618682



Azure Active Directory User

S-1-12-1-1414772360-7548652109-3974151294-7485145544



Active Directory User

S-1-5-21-415289841-218583201-7485910196-84512



Windows Logon ID

S-1-5-5-0-137426688

Access Token

What

An access token is an object that describes the security context of a process or thread.

Who

Access token mainly assigned to processes and threads. Access Token can be manipulated and queried by applications using the Windows API.

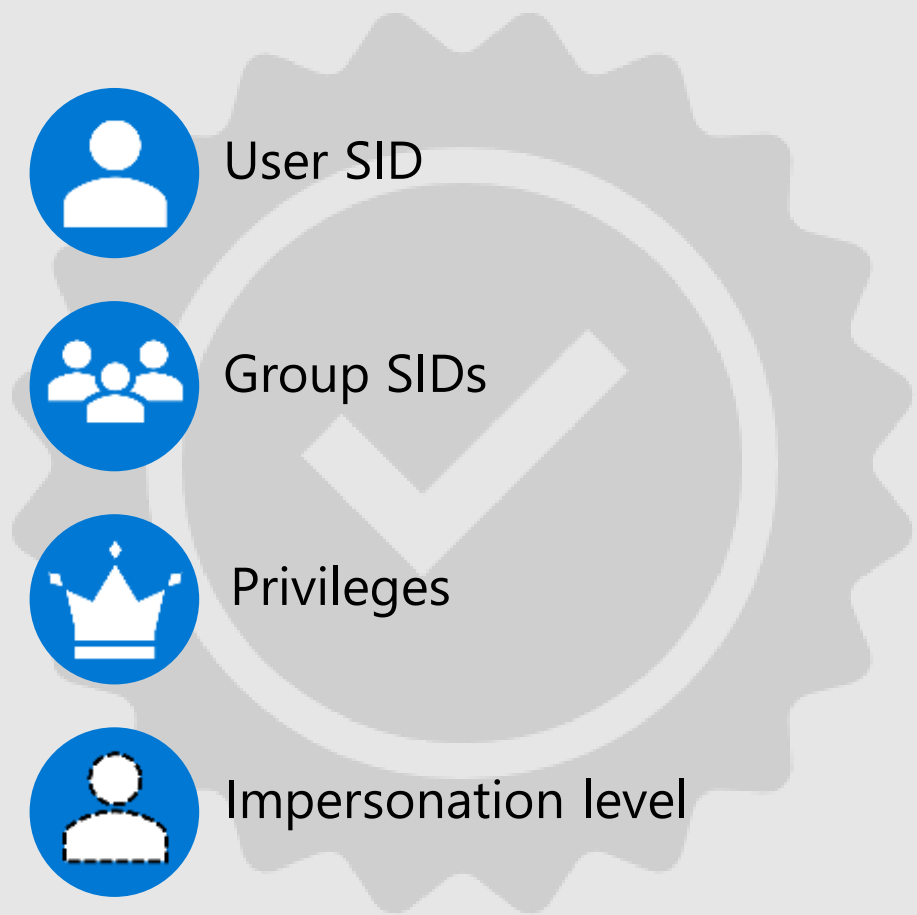
Where

Access Token are kernel objects. They exist in the kernel's address space and cannot be modified directly by applications. As with all kernel objects, applications are referencing tokens using handles. The Windows API provides many routines to query and modify tokens.

Why

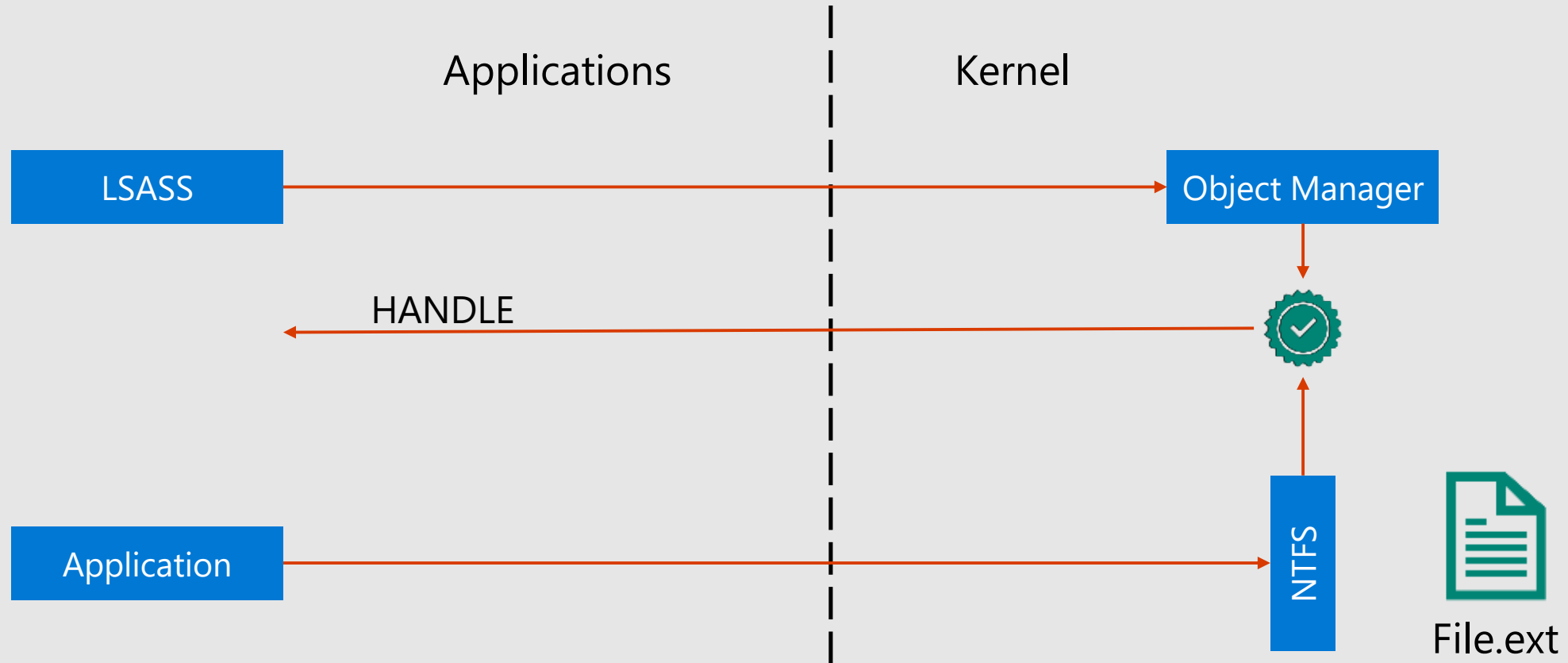
The system uses an access token to identify the user when a thread interacts with a securable object or tries to perform a system task that requires privileges.

Access Token



User
Group 1 SID
Group n SID
Privilege 1
Privilege n
Default Owner
Primary Group
Default Discretionary Access Control List (DACL)
Source
Type
Impersonation Level
Statistics
Restricting SID 1
Restricting SID n
TS Session ID
Session Reference
SandBox Inert
Audit Policy
Origin

Access token creation



Common Authorities

SID prefix	Authority
S-1-0-*	Null Authority
S-1-1-*	World Authority
S-1-2-*	Local Authority
S-1-5-*	NT Authority
S-1-11-*	Microsoft Account Authority
S-1-12-*	Azure AD Authority
S-1-15-*	Capability Authority
S-1-16-*	Mandatory Labels Authority
S-1-18-*	Authentication Authority

Wellknown SIDs

SID	Meaning
S-1-0-0	Null SID
S-1-1-0	Everyone
S-1-2-0	Local
S-1-2-1	Local Console
S-1-3-0	Creator Owner
S-1-3-1	Creator Group
S-1-3-4	Owner Rights

Wellknown SIDs in the NT Authority

SID	Meaning
S-1-5-2	Network session
S-1-5-3	Batch
S-1-5-4	Interactive
S-1-5-6	Service
S-1-5-7	Anonymous
S-1-5-9	Enterprise Domain Controllers
S-1-5-11	AuthenticatedUsers
S-1-5-14	Remote interactive session
S-1-5-15	This organization
S-1-5-21-*	Domain accounts
S-1-5-32-*	BUILTIN domain accounts
S-1-5-113	Local account

Wellknown SIDs in the BUILTIN domain

SID	Type	Account
S-1-5-32-544	A	BUILTIN\Administrators
S-1-5-32-545	A	BUILTIN\Users
S-1-5-32-546	A	BUILTIN\Guests
S-1-5-32-547	A	BUILTIN\Power Users
S-1-5-32-548	A	BUILTIN\Account Operators
S-1-5-32-549	A	BUILTIN\Server Operators
S-1-5-32-555	A	BUILTIN\Remote Desktop Users

Wellknown RIDs in domains

SID	Type	Account
S-1-5-21-x-y-z-500	U	Administrator
S-1-5-21-x-y-z-501	U	Guest
S-1-5-21-x-y-z-502	U	Krbtgt
S-1-5-21-x-y-z-512	G	Domain Admins
S-1-5-21-x-y-z-513	G	Domain Users
S-1-5-21-x-y-z-515	G	Domain Computers
S-1-5-21-x-y-z-516	G	Domain Controllers
S-1-5-21-x-y-z-519	G	Enterprise Admins

Integrity levels

Protected process integrity level	S-1-16-20480
System integrity level	S-1-16-16384
High integrity level	S-1-16-12288
Medium integrity level	S-1-16-8192
Low integrity level	S-1-16-4096
Untrusted integrity level	S-1-16-0

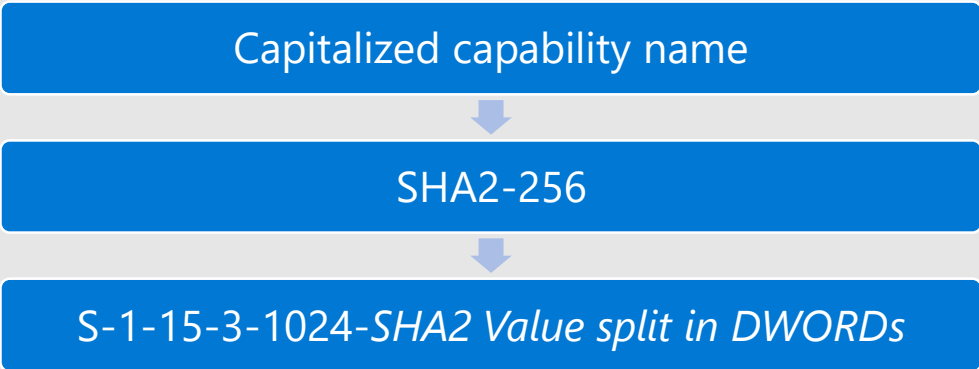


Capability SIDs

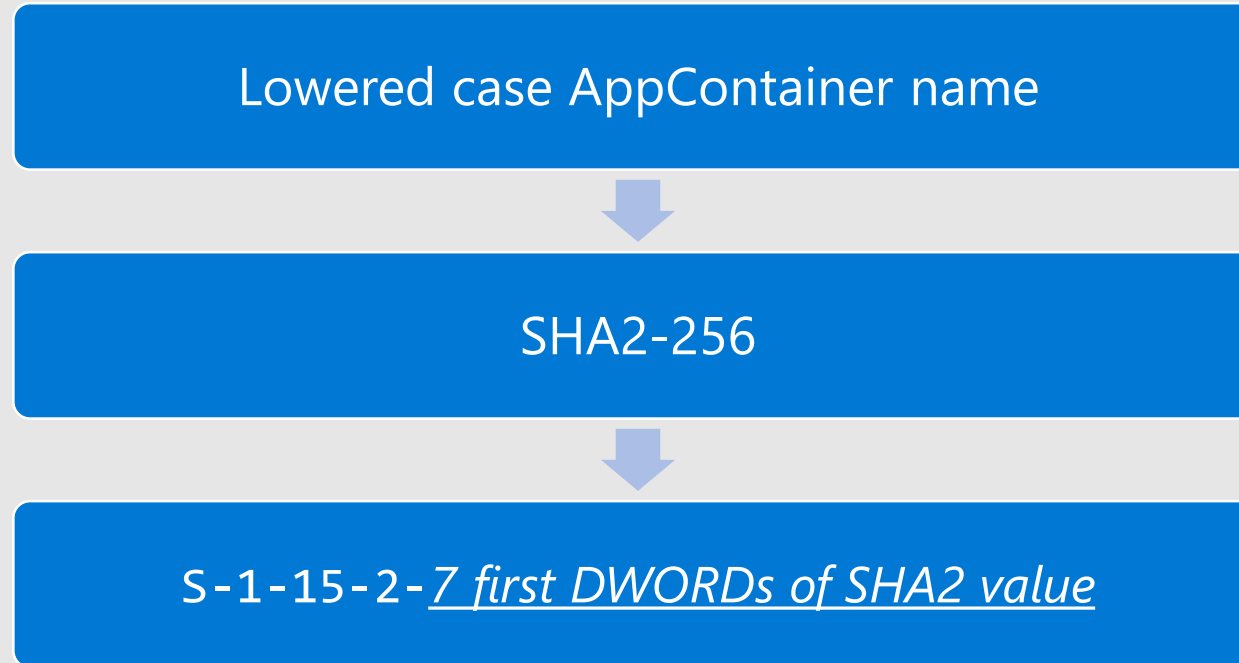
Legacy Capabilities

Capability	SID
INTERNETCLIENT	S-1-15-3-1
PICTURESLIBRARY	S-1-15-3-4
VIDEOSLIBRARY	S-1-15-3-5
ENTERPRISEAUTHENTICATION	S-1-15-3-8
REMOVABLESTORAGE	S-1-15-3-10

New Capabilities



Application Container SIDs



Special SIDs

- S-1-15-2-1 => ALL APPLICATION PACKAGES
- S-1-15-2-2 => ALL RESTRICTED APPLICATION PACKAGES

LocalSystem account

SID S-1-5-18

User Hive HKEY_USERS\DEFAULT ⇔ HKEY_USERS\S-1-5-18

Profile %WINDIR%\System32\config\systemprofile

Privileges

- SeAssignPrimaryTokenPrivilege
- SeLockMemoryPrivilege
- SeIncreaseQuotaPrivilege
- SeTcbPrivilege
- SeSecurityPrivilege
- SeTakeOwnershipPrivilege
- SeLoadDriverPrivilege
- SeSystemProfilePrivilege
- SeSystemtimePrivilege
- SeProfileSingleProcessPrivilege
- SeIncreaseBasePriorityPrivilege
- SeCreatePagefilePrivilege
- SeCreatePermanentPrivilege
- SeBackupPrivilege
- SeRestorePrivilege
- SeShutdownPrivilege
- SeDebugPrivilege
- SeAuditPrivilege
- SeSystemEnvironmentPrivilege
- SeChangeNotifyPrivilege
- SeUndockPrivilege
- SeManageVolumePrivilege
- SeImpersonatePrivilege
- SeCreateGlobalPrivilege
- SeIncreaseWorkingSetPrivilege
- SeTimeZonePrivilege
- SeCreateSymbolicLinkPrivilege
- SeDelegateSessionUserImpersonatePrivilege

Member of

- BUILTIN\Administrators

LUID

- 0x3e7

Mandatory Label

- System

Credential

- Kerberos : Yes
- NTLM : No by default

LocalService account

SID S-1-5-19

User Hive HKEY_USERS\S-1-5-19

Profile %WINDIR%\ServiceProfiles\LocalService

Privileges

- SeAssignPrimaryTokenPrivilege
- SeAuditPrivilege
- SeChangeNotifyPrivilege
- SeCreateGlobalPrivilege
- SeImpersonatePrivilege
- SeIncreaseQuotaPrivilege
- SeIncreaseWorkingSetPrivilege
- SeShutdownPrivilege
- SeSystemtimePrivilege
- SeTimeZonePrivilege
- SeUndockPrivilege

Member of

- BUILTIN\Users

LUID

- 0x3e5

Mandatory Label

- System

Credential

- Kerberos : No
- NTLM : No

NetworkService account

SID S-1-5-20

User Hive HKEY_USERS\S-1-5-20

Profile %WINDIR%\ServiceProfiles\NetworkService

Privileges

- SeAssignPrimaryTokenPrivilege
- SeAuditPrivilege
- SeChangeNotifyPrivilege
- SeCreateGlobalPrivilege
- SeImpersonatePrivilege
- SeIncreaseQuotaPrivilege
- SeShutdownPrivilege
- SeUndockPrivilege

Member of

- BUILTIN\Users

LUID

- 0x3e4

Mandatory Label

- System

Credential

- Kerberos : Yes
- NTLM : Yes

SID API

SID Formatting

- ConvertStringSidToSid
- ConvertSidToStringSid
- RtlLengthSidAsUnicodeString

SID Allocation

- RtlAllocateAndInitializeSid
- RtlInitializeSid
- RtlFreeSid
- RtlCreateServiceSid

SID Manipulation

- RtlIdentifierAuthoritySid
- RtlSubAuthoritySid
- RtlSubAuthorityCountSid
- RtlLengthSid
- RtlEqualSid
- RtlCopySid

Impersonation & Delegation



Privileges

- SeImpersonatePrivilege
- SeTcbPrivilege

Impersonation APIs

- ImpersonateAnonymousToken
- ImpersonateLoggedOnUser
- ImpersonateSelf
- RevertToSelf
- CreateProcessAsUser
- ImpersonateSecurityContext
- RevertSecurityContext
- RpcImpersonateClient
- RpcRevertToSelf
- RpcRevertToSelfEx

Additional API for Kerberos

- LsaLogonUser(KERB_S4U_LOGON)

SID – Additional resources

Well-known SIDs

<https://docs.microsoft.com/en-us/windows/win32/secauthz/well-known-sids>

SID API documentation

<https://docs.microsoft.com/en-us/windows/win32/secauthz/security-identifiers>

Token API

<https://docs.microsoft.com/en-us/windows/win32/secauthz/access-tokens>

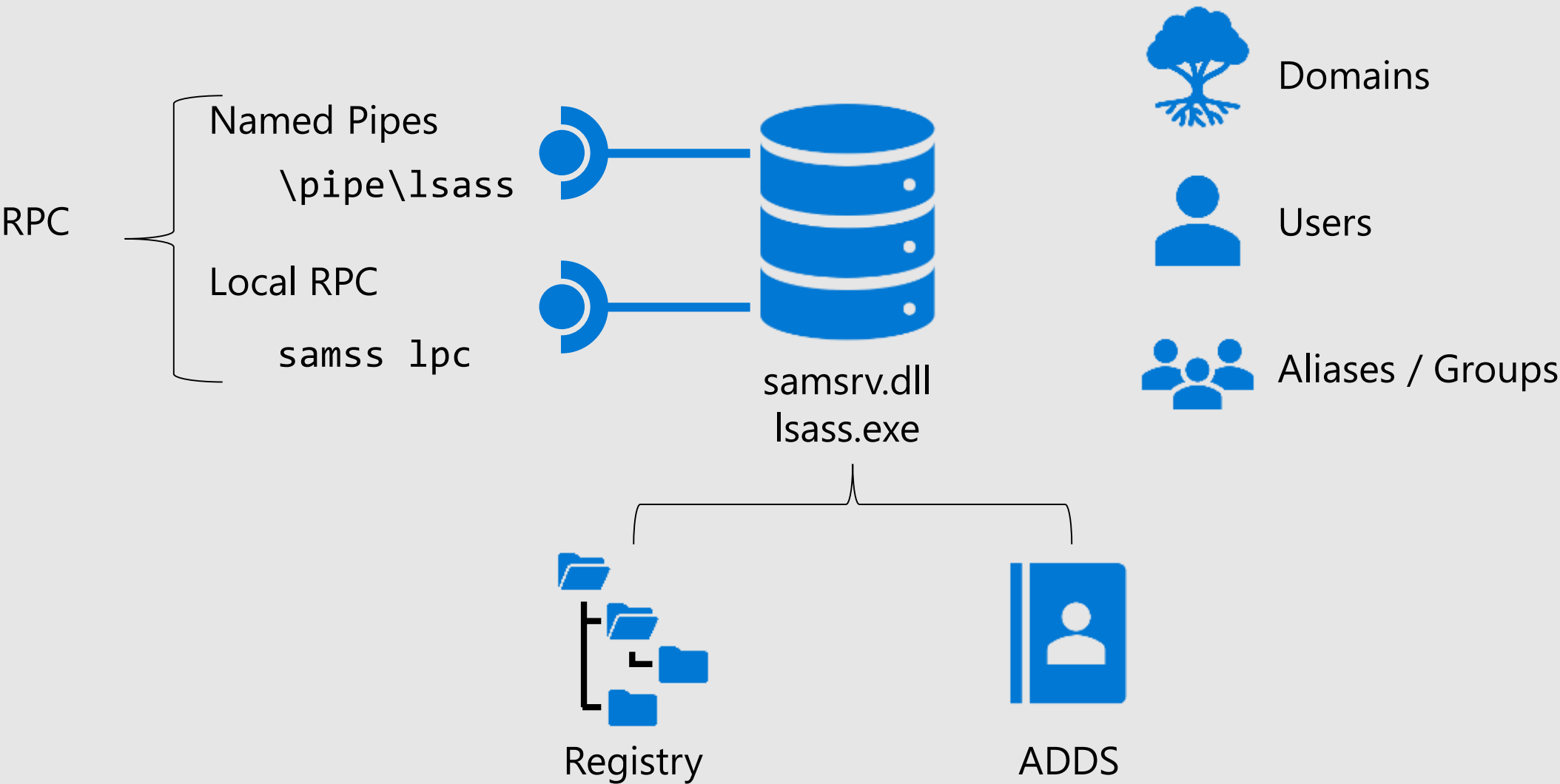
Chapter

1.1.2

Security Account Manager



SAM database



SAM Access Control

SAM & LSASS Services	<ul style="list-style-type: none">• Network access: Do not allow anonymous enumeration of SAM accounts and shares• Network access: Do not allow anonymous enumeration of SAM accounts• Network access: Allow anonymous SID/Name translation• Network access: Restrict clients allowed to make remote calls to SAM
RPC DACL	<ul style="list-style-type: none">• Network access: Let Everyone permissions apply to anonymous users
NP Transport	<ul style="list-style-type: none">• Network access: Named Pipes that can be accessed anonymously• Network access: Restrict anonymous access to Named Pipes and Shares

SAM/LSASS API

SAM Object Management

- SamConnect
- SamCloseHandle
- SamQueryInformationUser
- SamCreateUserInDomain
- SamSetInformationUser
- SamGetGroupsForUser
- SamDeleteUser
- SamEnumerateUsersInDomain

SAM Password Management

- SamChangePasswordUser

SAM Domain Management

- SamOpenDomain
- SamQueryInformationDomain
- SamEnumerateDomainsInSamServer
- SamSetInformationDomain

SAM Group Management

- SamAddMemberToAlias / SamAddMemberToGroup
- SamEnumerateGroupsInDomain
- SamDeleteAlias / SamDeleteGroup
- SamCreateAliasInDomain / SamCreateGroupInDomain
- SamOpenGroup
- SamQueryInformationGroup / SamQueryInformationAlias
- SamGetAliasMembership
- SamDeleteAlias / SamDeleteGroup
- SamRemoveMemberFromAlias

LSA Policy API

- LsaOpenPolicy / LsaClose
- LsaLookupSids
- LsaLookupNames
- LsaEnumerateTrustedDomains
- LsarQueryForestTrustInformation

SAM – Additional resources

SAM-R OpenSpecification

https://docs.microsoft.com/en-us/openspecs/windows_protocols/ms-samr/4df07fab-1bbc-452f-8e92-7853a3c7e380

LSA Policy reference

<https://docs.microsoft.com/en-us/windows/win32/secmgmt/using-lsa-policy>

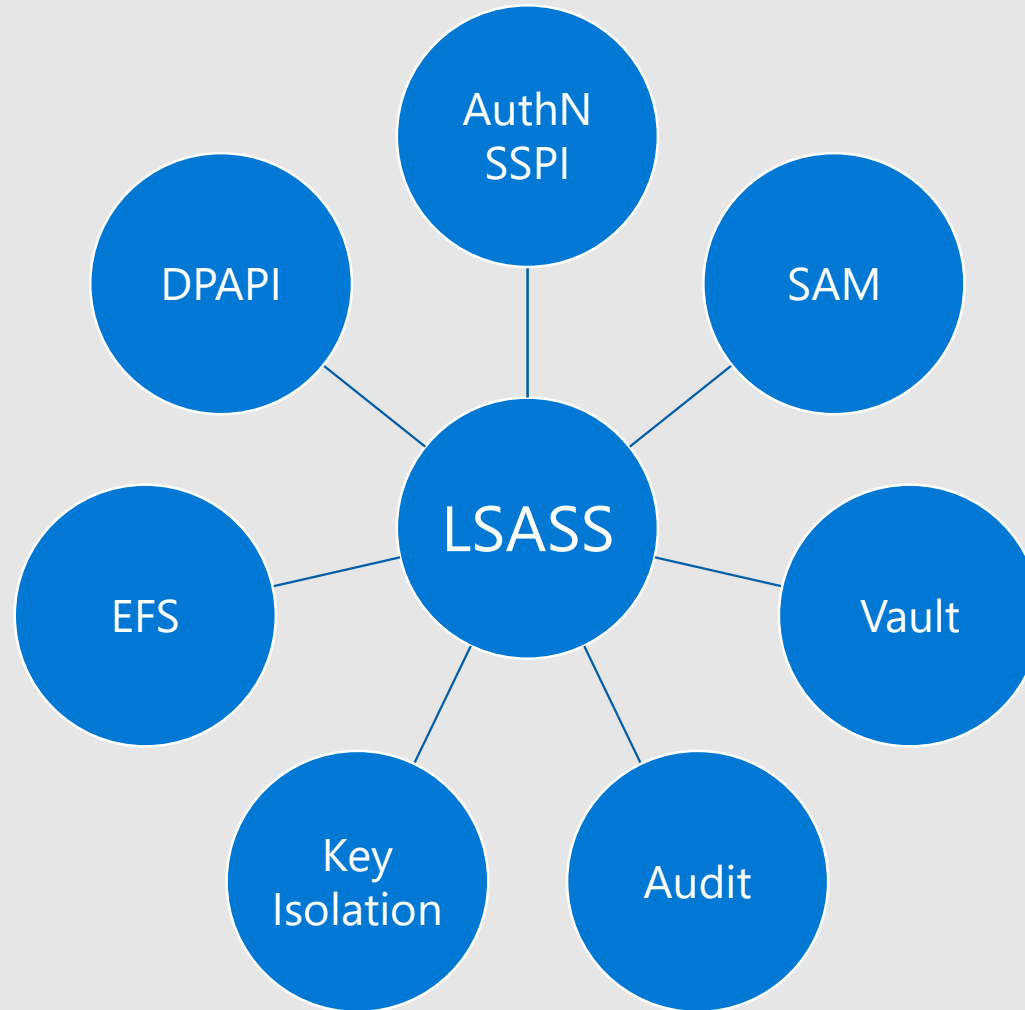
Chapter

1.1.3

**LSA sub-system
services**



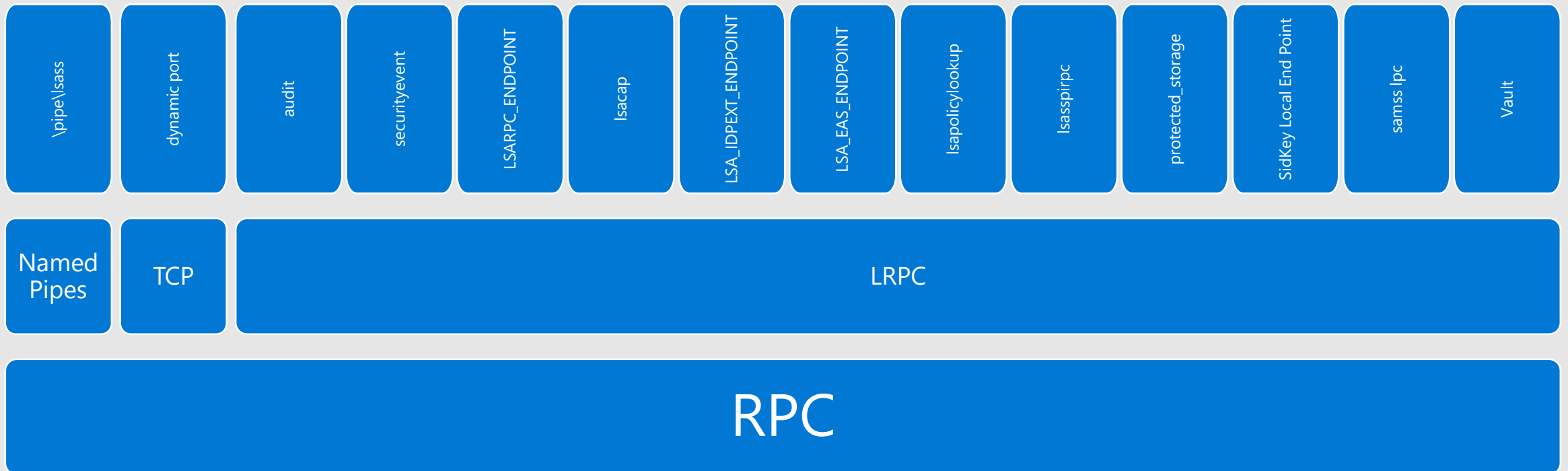
LSASS Services in brief



RPC Transports

LSASS services are reachable through RPC – *Remote Procedure Call*

RPC can be transported by many different protocols



RPC Security Configuration

Enable Endpoint Mapper Authentication

The screenshot shows the 'Enable RPC Endpoint Mapper Client Authentication' dialog box. At the top, the title bar reads 'Enable RPC Endpoint Mapper Client Authentication'. Below the title bar, there is a header section with the same title and two buttons: 'Previous Setting' and 'Next Setting'. The main content area has three radio buttons: 'Not Configured' (selected), 'Enabled', and 'Disabled'. To the right of these buttons is a 'Comment:' text box. Below the radio buttons is a 'Supported on:' dropdown menu showing 'At least Windows XP Professional with SP2'. At the bottom, there are 'Options:' and 'Help:' sections. The 'Help' section contains a detailed explanation of the policy setting and its impact on RPC clients and servers. At the very bottom, there are 'OK', 'Cancel', and 'Apply' buttons.

Enable RPC Endpoint Mapper Client Authentication

Previous Setting Next Setting

☒ Not Configured ☐ Enabled ☐ Disabled

Comment:

Supported on: At least Windows XP Professional with SP2

Options:

Help:

This policy setting controls whether RPC clients authenticate with the Endpoint Mapper Service when the call they are making contains authentication information. The Endpoint Mapper Service on computers running Windows NT4 (all service packs) cannot process authentication information supplied in this manner.

If you disable this policy setting, RPC clients will not authenticate to the Endpoint Mapper Service, but they will be able to communicate with the Endpoint Mapper Service on Windows NT4 Server.

If you enable this policy setting, RPC clients will authenticate to the Endpoint Mapper Service for calls that contain authentication information. Clients making such calls will not be able to communicate with the Windows NT4 Server Endpoint Mapper Service.

If you do not configure this policy setting, it remains disabled. RPC clients will not authenticate to the Endpoint Mapper Service, but they will be able to communicate with the Windows NT4

OK Cancel Apply

Force client authentication

The screenshot shows the 'Restrict Unauthenticated RPC clients' dialog box. At the top, the title bar reads 'Restrict Unauthenticated RPC clients'. Below the title bar, there is a header section with the same title and two buttons: 'Previous Setting' and 'Next Setting'. The main content area has three radio buttons: 'Not Configured' (selected), 'Enabled', and 'Disabled'. To the right of these buttons is a 'Comment:' text box. Below the radio buttons is a 'Supported on:' dropdown menu showing 'At least Windows XP Professional with SP2'. At the bottom, there are 'Options:' and 'Help:' sections. The 'Options:' section has a dropdown menu labeled 'RPC Runtime Unauthenticated Client Restriction to Apply:'. The 'Help' section contains a detailed explanation of the policy setting and its impact on RPC servers and clients. At the very bottom, there are 'OK', 'Cancel', and 'Apply' buttons.

Restrict Unauthenticated RPC clients

Previous Setting Next Setting

☒ Not Configured ☐ Enabled ☐ Disabled

Comment:

Supported on: At least Windows XP Professional with SP2

Options:

Help:

RPC Runtime Unauthenticated Client Restriction to Apply:

This policy setting controls how the RPC server runtime handles unauthenticated RPC clients connecting to RPC servers.

This policy setting impacts all RPC applications. In a domain environment this policy setting should be used with caution as it can impact a wide range of functionality including group policy processing itself. Reverting a change to this policy setting can require manual intervention on each affected machine. This policy setting should never be applied to a domain controller.

If you disable this policy setting, the RPC server runtime uses the value of "Authenticated" on Windows Client, and the value of "None" on Windows Server versions that support this policy setting.

If you do not configure this policy setting, it remains disabled. The RPC server runtime will behave as though it was enabled with the value of "Authenticated" used for Windows Client and the value of "None" used for Server SKUs that support this policy setting.

OK Cancel Apply

Notable RPC interface GUIDs

12345778-1234-abcd-ef00-0123456789ab

- LSA Policy
- Credential Manager

12345778-1234-abcd-ef00-0123456789ac

- SAM

c0d930f0-b787-4124-99bc-21f0ecb642ce

- Internet Accounts

c681d488-d850-11d0-8c52-00c04fd90f7e

- Encrypted File System

4f32adc8-6052-4a04-8701-293ccf2096f0

- SSPI

11220835-5b26-4d94-ae86-c3e475a809de

- DPAPI

7f1317a8-4dea-4fa2-a551-df5516ff8879

- DPAPIng

b25a52bf-e5dd-4f4a-aea6-8ca7272a0e86

- Key Isolation

bb8b98e8-84dd-45e7-9f34-c3fb6155eed

- Vault

8fb74744-b2ff-4c00-be0d-9ef9a191fe1b

- Windows Hello for Business PopKey service

Popular 3rd party tools to assess RPC configuration

Nmap msrpc-enum

<https://nmap.org/nsedoc/scripts/msrpc-enum.html>

Impacket python library

<https://www.secureauth.com/labs/open-source-tools/impacket/>

Chapter

1.1.4

Session Lifecycle



What is a session ?

Session

- Kernel semantic
- Way to partition the system between users.
- Partition is enforced in kernel and user mode
- Session 0, Session 1, Session 2..n
- Orchestrator : Winlogon.exe

Logon Session

- LSASS semantic
- Created when authenticating credentials
- Identified with a LUID (64bit locally unique identifier)

Winlogon Initialization

Sessions are created

- Ahead of time for local sessions (ie: sessions connected using a monitor, mouse and keyboard)
- On-demand for remote sessions (using RDP)

Smss.exe creates a new session space in the Kernel and associated graphical objects

- The most notable object is the first Desktop object so that the UI can actually paint graphics on the screen

Smss.exe then starts winlogon.exe

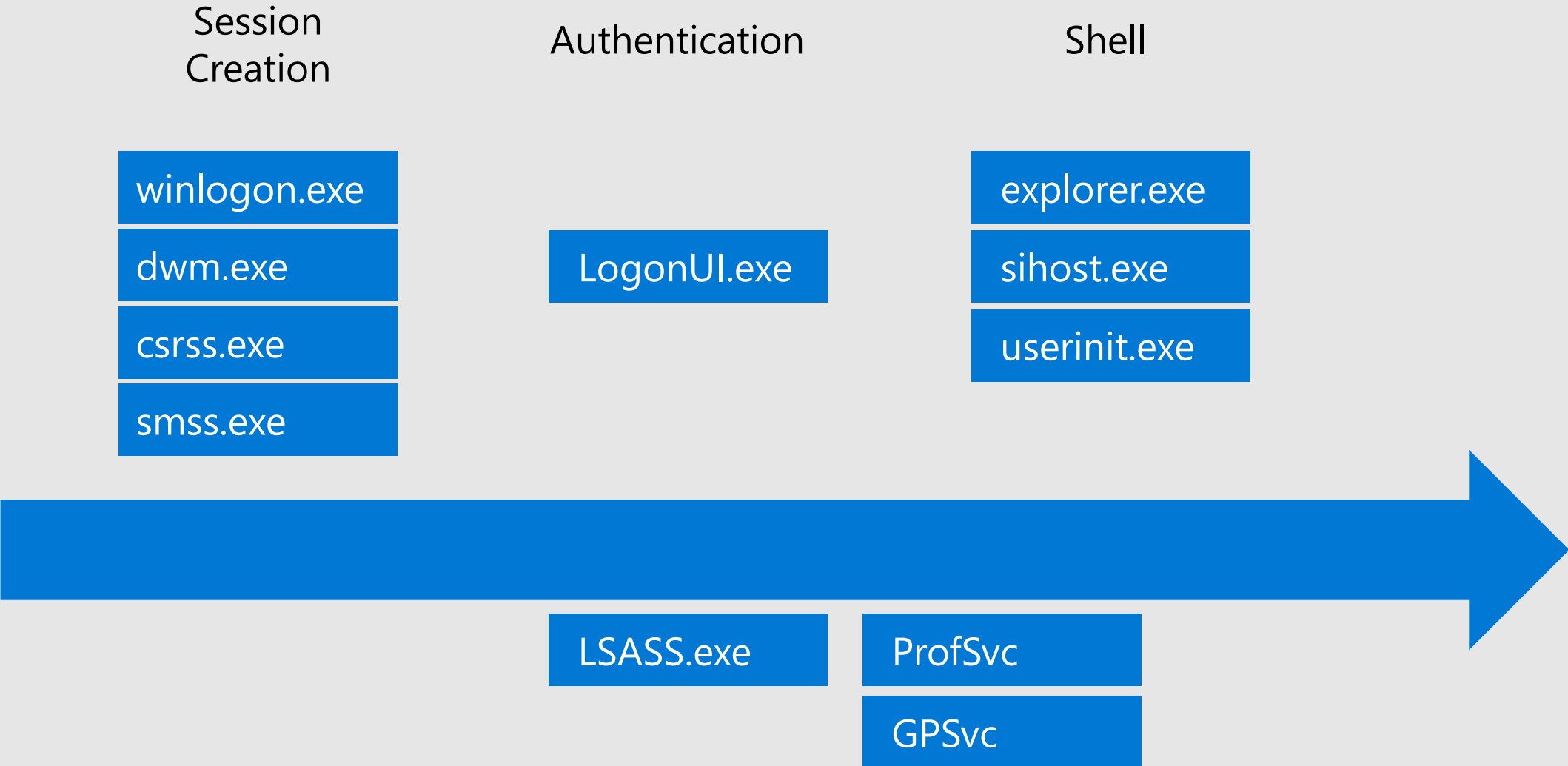
- Winlogon.exe is the wininit.exe counter part for user sessions
- Winlogon.exe is responsible for managing the session lifetime
- Winlogon.exe stays running until session is completely logged off

Winlogon Initialization

Winlogon.exe

- Creates and opens an interactive window station
`\Sessions\1\Windows\WindowStations\WinSta0`
- Creates and opens 2 Desktop objects
`\Sessions\1\Windows\WindowStations\WinSta0\Default`
`\Sessions\1\Windows\WindowStations\WinSta0\Winlogon`
- The Winlogon desktop is also called the secure desktop as this is where UI picks user's credential
- Desktop are securable objects like any Kernel object. Only SYSTEM account can access the Winlogon desktop.

User session - creation process



CredentialProviders

- COM Objects – Component Object Model
- ICredentialProvider
- ICredentialProviderCredential
- ICredentialProviderFilter

Winlogon Notification Packages

Third-party application can register to be notified on winlogon state changes

Entry must be created under

HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\Winlogon\Notify

Possible events to be notified on:

- Lock, Unlock

- Logoff, Logon

- Startup, Shutdown

- StartScreenSaver, StopScreenSaver

- StartShell

User session – Additional resources

Credential providers

<https://docs.microsoft.com/en-us/windows/win32/secauthn/credential-providers-in-windows>

ICredentialProvider interface

<https://docs.microsoft.com/en-us/windows/win32/api/credentialprovider/nncredentialprovider-icredentialprovider>

Chapter

1.1.5

Authentication packages



Authentication in Windows

Short story of authentication in Windows



Authentication is handled by Authentication Packages

- DLLs loaded by LSASS.exe
- 3rd-party AP are supported with special code signing requirements
- Benefit: Credential storage and derivation happen in a separate process

Authentication API entry points

Entry point for logging in users is the LsaLogonUser() API.

Prior to call this API, Logon applications must register using

LsaRegisterLogonProcess()

Applications performing sensitive Authentication tasks like delegation must use this API. Caller must be granted SeTcbPrivilege.

LsaConnectUntrusted()

For applications performing non-sensitive authentication tasks like signing in users with passwords, querying package properties or listing installed packages

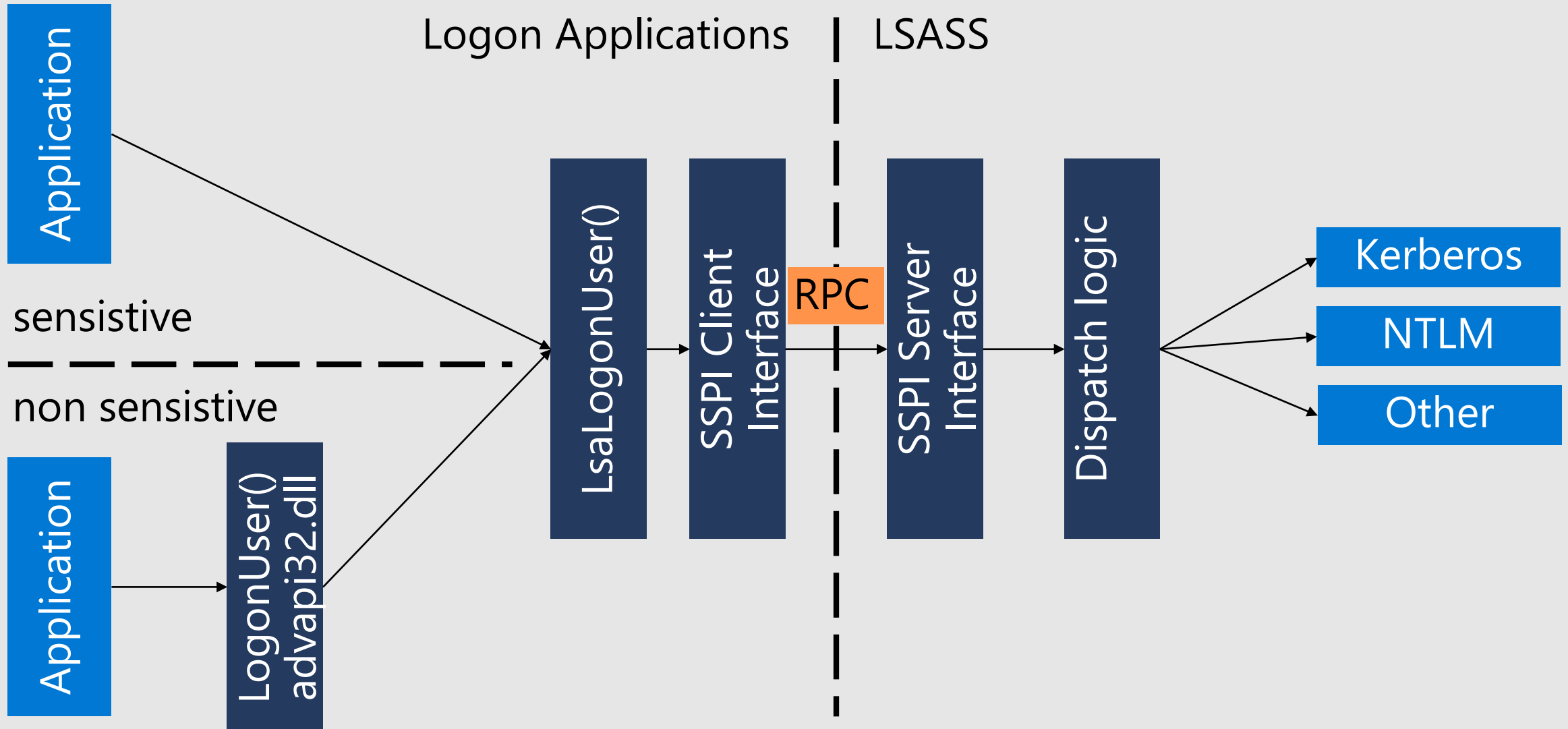
Authentication API

Windows provides some helper API for applications

Basically, a wrapper around LsaLogonUser() hiding the application registration process

Function	Description
<u>LogonUser</u>	Attempts to log a user on to the local computer.
<u>LogonUserEx</u>	Attempts to log a user on to the local computer. This function is an extended version of the <u>LogonUser</u> function and retrieves information about the logged-on user's <u>security identifier</u> (SID), profile, and quota limits.
<u>LogonUserExExW</u>	The <u>LogonUserExExW</u> function attempts to log a user on to the local computer.

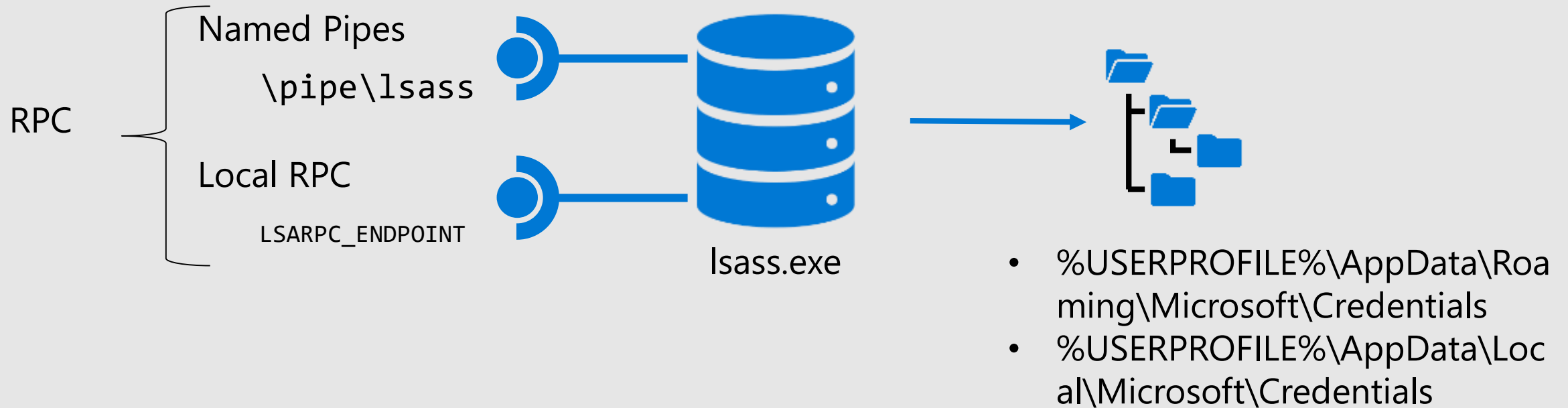
Authentication flow



Authentication Packages

Name	Use
Negotiate	Implements SPNego RFC4178
NegoExtender	Negotiate + Extensions
Kerberos	Kerberos v5
NTLM	NTLM authentication protocol
TSSSP	CredSSP. Forwards credentials to RDP hosts
pku2u	Peer to peer user certificate authentication
CloudAP	Cloud identity providers (like AAD)
WDigest	Digest protocol as in RFCs 2617 and 2831
Schannel	Implementation of SSL and TLS
Microsoft Unified Security Protocol Provider	Same as Schannel
MICROSOFT_AUTHENTICATION_PACKAGE_V1_0	Same as NTLM

Credential Manager



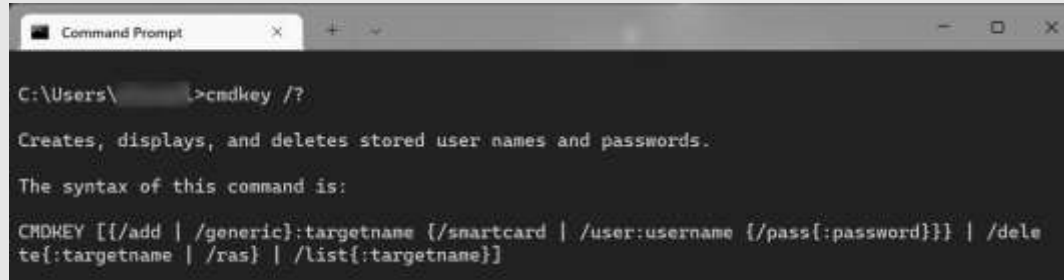
Credential Manager API

- CredWriteW
- CredReadW
- CredFree
- CredRenameW
- CredEnumerateW
- CredFindBestCredentialW

```
typedef struct _CREDENTIALA {  
    DWORD                Flags;  
    DWORD                Type;  
    LPSTR                TargetName;  
    LPSTR                Comment;  
    FILETIME             LastWritten;  
    DWORD                CredentialBlobSize;  
    LPBYTE               CredentialBlob;  
    DWORD                Persist;  
    DWORD                AttributeCount;  
    PCREDENTIAL_ATTRIBUTEA Attributes;  
    LPSTR                TargetAlias;  
    LPSTR                UserName;  
} CREDENTIALA, *PCREDENTIALA;
```

Credential Manager tools

Command line : cmdkey.exe



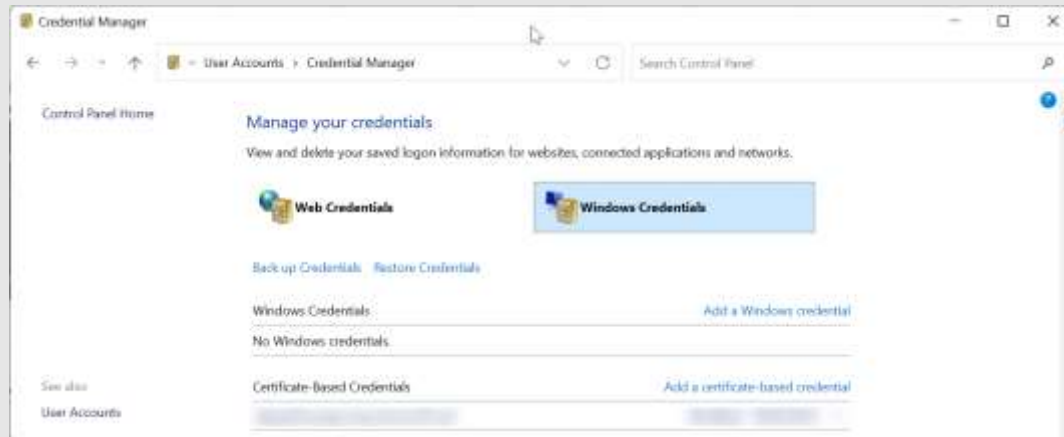
```
Command Prompt
C:\Users\>cmdkey /?

Creates, displays, and deletes stored user names and passwords.

The syntax of this command is:

CMDKEY [[/add | /generic]:targetname [/smartcard | /user:username [/pass[:password]]] | /delete[:targetname | /ras] | /list[:targetname]]
```

Graphical UI: Control Panel



Credential Manager – Additional resources

CredWrite API

<https://docs.microsoft.com/en-us/windows/win32/api/wincred/nf-wincred-credwritew>

CredRead API

<https://docs.microsoft.com/en-us/windows/win32/api/wincred/nf-wincred-credreadw>

Chapter

1.1.6

**SSPI - Security
Support Provider
Interface**



SSPI Terminology

AP – Authentication Package: DLL - Handles Authentication request

SSP – Security Support Package: DLL – Provides support for a given security protocol

SSPI – Security Support Provider Interface: API to access services from SSP/AP

Credential: Identity of a peer

Context: Security association between 2 peers

SSPI call dispatch

- Logon applications call a generic interface provided by LSASS.
- LSASS internally dispatch the call to the correct package based on package name or ID
- Packages can be an AP, an SSP or both. Kerberos, for instance, is both and AP and an SSP. We call it an SSP/AP.
- Benefit: Authentication mechanics and secrets storage occur in a different process.



SSPI Functions Classes

Package Management

Functions that list the available security packages and select a package.

Credential Management

Functions that create and work with handles to the credentials of principals.

Context Management

Functions that use credentials handles to create a security context.

Message Support

Functions that use security contexts to ensure message integrity and privacy during message exchanges over the secured connection. Integrity is achieved through message signing and signature verification. Privacy is achieved through message encryption and decryption.

SSPI API

Authentication Helpers

- LogonUser
- LogonUserEx / LogonUserExEx

Package Management

- EnumerateSecuritypackages
- InitSecurityInterface
- QuerySecurityPackageInfo

Credential Management

- AcquireCredentialsHandle
- FreeCredentialsHandle
- QueryCredentialsAttributes

Context Management

- AcceptSecurityContext
- InitializeSecurityContext
- DeleteSecurityContext
- ImpersonateSecurityContext
- QueryContextAttributes
- SetContextAttributes
- QuerySecurityContextToken
- CompleteAuthToken

Message

- DecryptMessage
- EncryptMessage
- MakeSignature
- VerifySignature

Package Management

SSPI package management functions initiate a security package, enumerate available packages, and query the attributes of a security package.

The following SSPI functions provide management services for security packages.

Function	Description
EnumerateSecurityPackages	Lists available security packages and their capabilities.
InitSecurityInterface	Retrieves a pointer to a security support provider (SSP) dispatch table.
QuerySecurityPackageInfo	Retrieves information about a specified security package . This information includes the bounds on sizes of authentication information, credentials , and contexts.

Credential Management APIs

SSPI credential management functions provide a credentials handle, a reference to an opaque security object, for accessing a principal. The security object is opaque because the application has access only to the handle and not to the actual contents of the structure.

A credential handle is a 64-bit value between {0x00000000, 0x00000000} and {0xFFFFFFFF, 0xFFFFFFFFFE}

Function	Description
<u>AcquireCredentialsHandle (General)</u>	Acquires a handle to the preexisting credentials of a specified principal.
<u>ExportSecurityContext</u>	Exports a security context into a context buffer.
<u>FreeCredentialsHandle</u>	Releases a credential handle and associated resources.
<u>ImportSecurityContext</u>	Imports a security context exported by using <u>ExportSecurityContext</u> into the current process.
<u>QueryCredentialsAttributes</u>	Retrieves the attributes of a credential, such as the name associated with the credential.

Context Management APIs

In a communication link, the client and server cooperate to create a shared security context. The client and server both use the security context with message support functions to ensure message integrity and privacy during the connection.

Security contexts are opaque security objects. Information in the security context is not available to the application. Context management functions create and use context handles and the security package dereferences the context handle to access its security content.

A context handle is a 64-bit value between {0x00000000, 0x00000000} and {0xFFFFFFFF, 0xFFFFFFFFE}.

Function	Description
<u>AcceptSecurityContext</u>	Used by a server to create a security context based on an opaque message received from a client.
<u>ImpersonateSecurityContext</u>	Impersonates the security context to appear as the client to the system.
<u>CompleteAuthToken</u>	Completes an authentication token. This function is used by protocols, such as DCE, that need to revise the security information after the transport application has updated some message parameters.

Context Management APIs

Function	Description
<u>DeleteSecurityContext</u>	Frees a security context and associated resources.
<u>InitializeSecurityContext (General)</u>	Used by a client to initiate a security context by generating an opaque message to be passed to a server.
<u>QueryContextAttributes (General)</u>	Enables a transport application to query a security package for certain attributes of a security context.
<u>QuerySecurityContextToken</u>	Obtains the access token for a client security context and uses it directly.
<u>SetContextAttributes</u>	Enables a transport application to set attributes of a security context for a security package. This function is supported only by the Schannel security package.

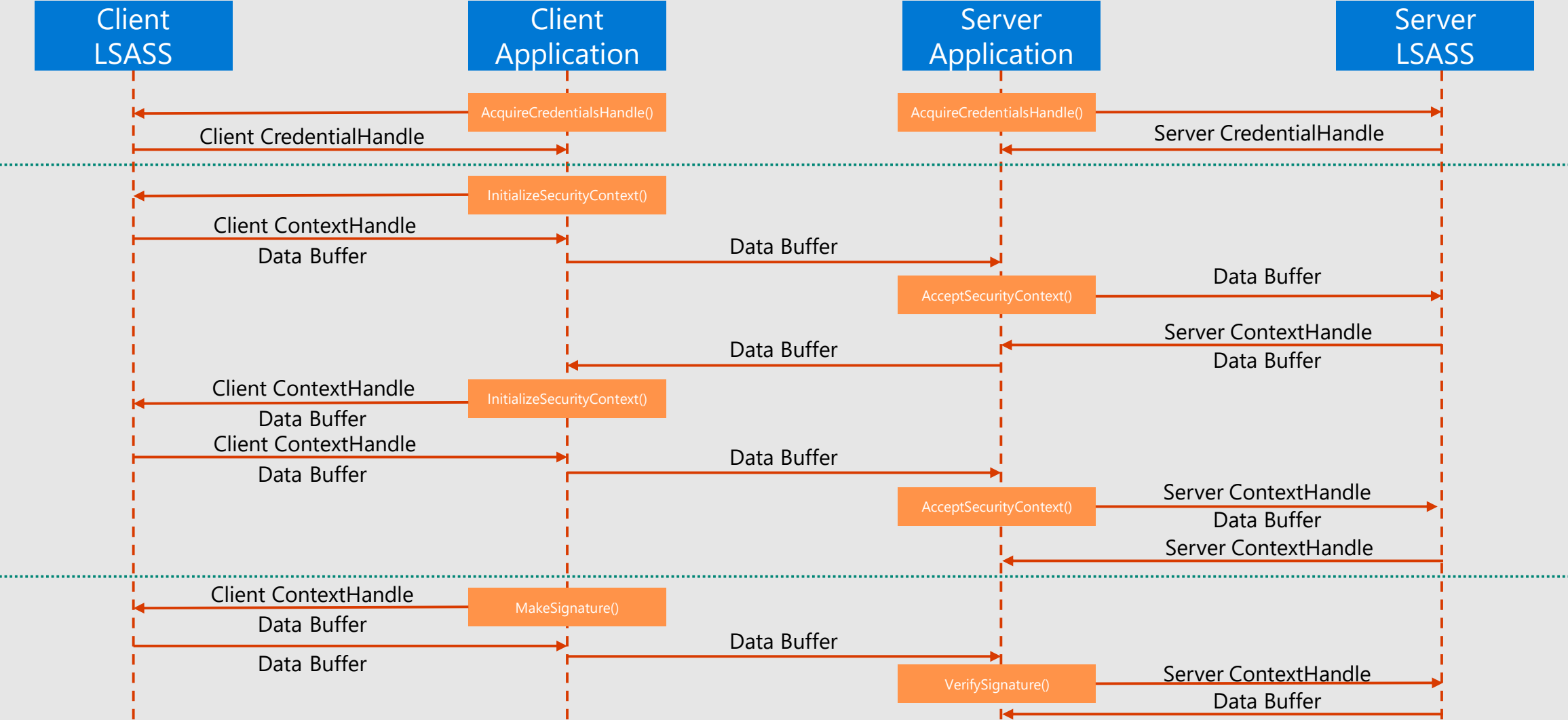
SSPI message support functions enable an application to transmit and receive tamper-resistant messages and to encrypt and decrypt messages.

These functions work with one or more buffers that contain a message and with a security context created by the context management functions.

The functions' behavior differs based on whether a connection, datagram, or stream context is in use.

Function	Description
<u>DecryptMessage (General)</u>	Decrypts an encrypted message by using the session key from a security context.
<u>EncryptMessage (General)</u>	Encrypts a message by using the session key from a security context.
<u>MakeSignature</u>	Generates a cryptographic checksum of the message, and also includes sequencing information to prevent message loss or insertion.
<u>VerifySignature</u>	Verifies the signature of a message received that was signed by the sender by using the <u>MakeSignature</u> function.

SSPI Processing



SSPI – Additional resources

SSPI Documentation

<https://docs.microsoft.com/en-us/windows/win32/secauthn/sspi>

Chapter

1.1.7

**UAC – User Account
Control**

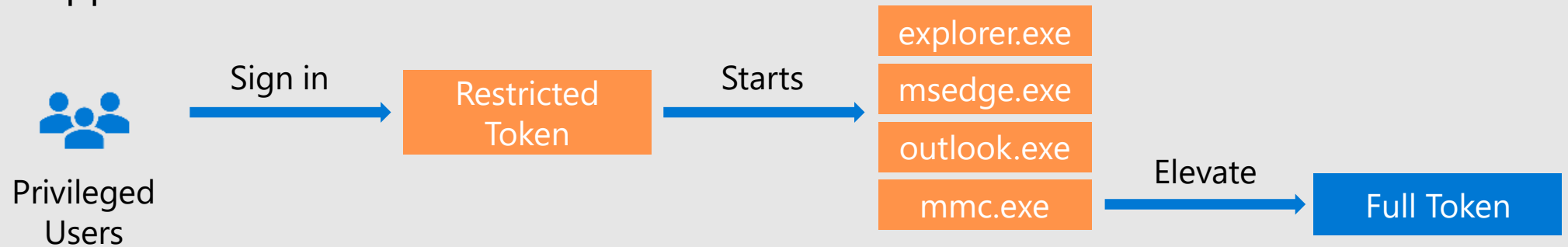


UAC – User Account Control

Usual Admin login



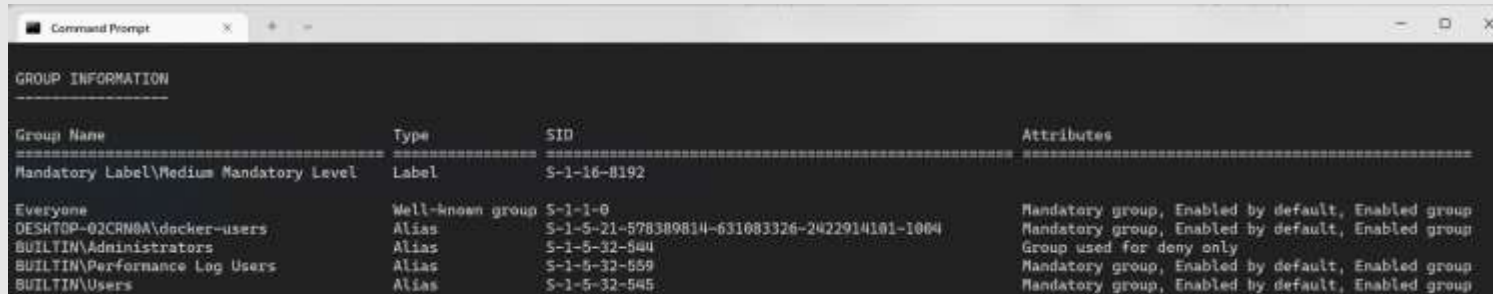
Admin Approval mode



Benefits of UAC

- Disables privileges and groups when not required
- Approval on secure desktop
- Checks Authenticode signature before elevation
- File and registry virtualization

Filtered groups



Group Name	Type	SID	Attributes
Mandatory Label\Medium Mandatory Level	Label	S-1-16-8192	
Everyone	Well-known group	S-1-1-0	Mandatory group, Enabled by default, Enabled group
DESKTOP-02CRN0A\docker-users	Alias	S-1-5-21-578309814-631083326-2422914101-1004	Mandatory group, Enabled by default, Enabled group
BUILTIN\Administrators	Alias	S-1-5-32-544	Group used for deny only
BUILTIN\Performance Log Users	Alias	S-1-5-32-559	Mandatory group, Enabled by default, Enabled group
BUILTIN\Users	Alias	S-1-5-32-545	Mandatory group, Enabled by default, Enabled group

Groups

- Domain Admins
- Read-only Domain Controllers
- Enterprise Read-only Domain Controllers
- Cert Publishers
- Schema Admins
- Enterprise Admins
- Group Policy Creator Owners
- RAS and IAS Servers
- Administrators
- Power Users
- Account Operators
- Server Operators
- Print Operators
- Backup Operators
- Pre-Windows 2000 Compatible Access
- Network Configuration Operators
- Cryptographic Operators
- NT AUTHORITY\Local account and member of Administrators group

UAC Settings

Admin Approval mode	<ul style="list-style-type: none">• User Account Control: Admin Approval Mode for the Built-in Administrator account• User Account Control: Turn on Admin Approval Mode
Prompt behavior	<ul style="list-style-type: none">• User Account Control: Behavior of the elevation prompt for administrators in Admin Approval Mode• User Account Control: Behavior of the elevation prompt for standard users
Compatibility	<ul style="list-style-type: none">• User Account Control: Detect application installations and prompt for elevation• User Account Control: Virtualize file and registry write failures to per-user locations
Additional Security	<ul style="list-style-type: none">• User Account Control: Switch to the secure desktop when prompting for elevation• User Account Control: Only elevate executable files that are signed and validated
Accessibility	<ul style="list-style-type: none">• User Account Control: Allow UIAccess application to prompt for elevation without using the secure desktop• User Account Control: Only elevate UIAccess applications that are installed in secure locations

UAC – Additional resources

How UAC works

<https://docs.microsoft.com/en-us/windows/security/identity-protection/user-account-control/how-user-account-control-works>

User Account Control security policy settings

<https://docs.microsoft.com/en-us/windows/security/identity-protection/user-account-control/user-account-control-security-policy-settings>

Chapter

1.1.8

Service accounts

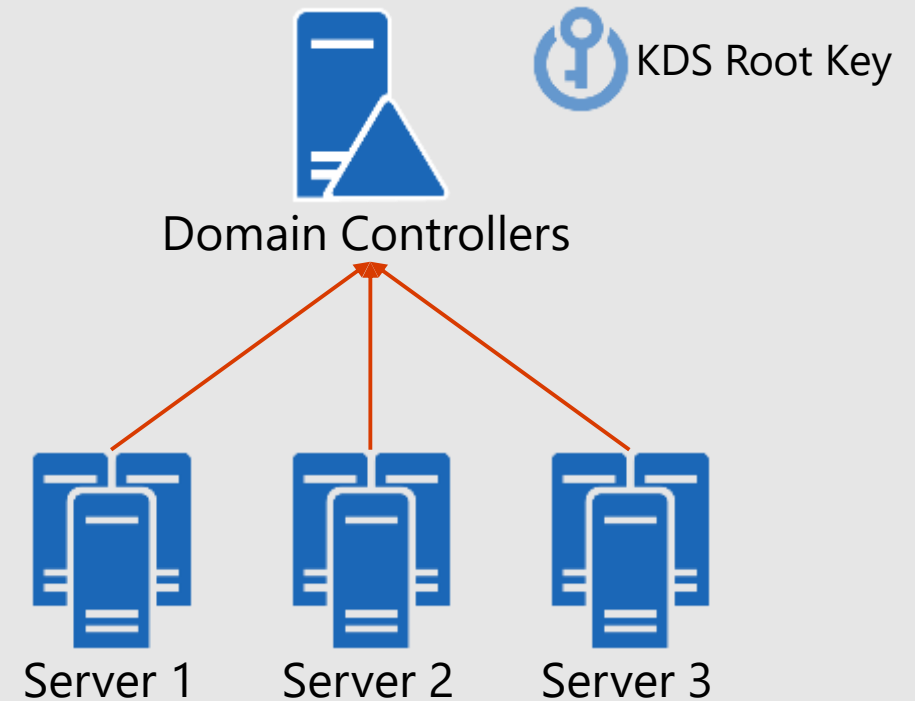


Risks inherent to service accounts

- Usually regular user accounts. Service nature is a convention.
- Password management
- Requires 3rd party password sharing solution

GMSA Operation

- Still password credential
- Password managed by allowed servers
- Explicit list of servers allowed to host the gMSA
- Uses KDS to share the password



GMSA Cmd-lets

Powershell

```
Add-KdsRootKey -EffectiveTime ((get-date).AddHours(-10));
```

```
New-ADServiceAccount ITFarm1  
-DNSHostName ITFarm1.contoso.com  
-PrincipalsAllowedToRetrieveManagedPassword ITFarmHosts$  
-KerberosEncryptionType AES128, AES256  
-ServicePrincipalNames http/ITFarm1.contoso.com/contoso.com
```

gMSA – Additional resources

GMSA Operations

<https://docs.microsoft.com/en-us/windows-server/security/group-managed-service-accounts/getting-started-with-group-managed-service-accounts>

Configuring delegation

<https://docs.microsoft.com/en-us/windows-server/security/group-managed-service-accounts/configure-kerberos-delegation-group-managed-service-accounts>

