[MS-KILE]:
Kerberos Protocol Extensions

Revision Summary

Date	Revision History	Revision Class	Comments
10/22/2006	0.01		MCPP Milestone 1 Initial Availability
01/19/2007	1.0		MCPP Milestone 1
03/02/2007	1.1		Monthly release
04/03/2007	1.2		Monthly release
05/11/2007	1.3		Monthly release

Date	Revision History	Revision Class	Comments
06/01/2007	1.3.1	Editorial	Revised and edited the technical content.
07/03/2007	2.0	Major	Revised technical content in several sections and created two new sections.
07/20/2007	2.0.1	Editorial	Revised and edited the technical content.
08/10/2007	3.0	Major	Updated content based on feedback.
09/28/2007	3.1	Minor	Made technical and editorial changes based on feedback.
10/23/2007	3.2	Minor	Made technical and editorial changes based on feedback.
11/30/2007	3.3	Minor	Made technical and editorial changes based on feedback.
01/25/2008	3.3.1	Editorial	Revised and edited the technical content.

Table of Contents

1 Introduction

Kerberos Protocol Extensions (KILE) specifies extensions to the Kerberos Network Authentication Service (V5) protocol [RFC4120]. These extensions provide additional capability for authorization information including group memberships, interactive logon information and integrity levels as well as constrained delegation and encryption supported by **Kerberos principals**.

Note Throughout the remainder of this specification the Kerberos Network Authentication Service (V5) protocol will be referred to simply as Kerberos V5.

1.1 Glossary

The following terms are defined in [MS-GLOS]:

Active Directory (AD)
AP Exchange
AS Exchange
Authentication Service (AS)
Authenticator
Authorization Data
Directory
Domain
Generic Security Services (GSS)
Kerberos Principal
Key
Key Distribution Center (KDC)
KRB_AP_REQ/KRB_AP_REP
KRB_AS_REQ/KRB_AS_REP
KRB_PRIV Exchange
KRB_SAFE Exchange
Object Identifier (OID)
Pre-Authentication
Privilege Attribute Certificate (PAC)
Realm
Secret Key
Security Support Provider Interface (SSPI)
Service
Service Principal
Service Principal Name (SPN)
Service Ticket
Session
Session Key
SRV Record
TGS Exchange
Ticket
Ticket-Granting Service (TGS)
Ticket-Granting Ticket (TGT)

The following terms are specific to this document:

Context Session Key: A variant of a cryptographic key used in the generation and processing of per-message tokens that uses the Kerberos session key directly ([RFC1964] section 1.2).

Integrity Level: The attributed trustworthiness of an entity or object.

Security Package: The software implementation of a security protocol. Security packages are contained in security support provider components or security support provider/authentication package components.

Ticket Session Key: The **session key** within a **ticket**.

MAY, SHOULD, MUST, SHOULD NOT, MUST NOT: These terms (in all caps) are used as described in [RFC2119]. All statements of optional behavior use either MAY, SHOULD, or SHOULD NOT.

1.2 References

1.2.1 Normative References

We conduct frequent surveys of the normative references to assure their continued availability. If you have any issue with finding a normative reference, please contact dochelp@microsoft.com. We will assist you in finding the relevant information. Please check the archive site, http://msdn2.microsoft.com/en-us/library/E4BD6494-06AD-4aed-9823-445E921C9624, as an additional source.

[C706] The Open Group, "DCE 1.1: Remote Procedure Call", C706, August 1997, http://www.opengroup.org/public/pubs/catalog/c706.htm

[FIPS140] National Institute of Standards and Technology, "Federal Information Processing Standards Publication 140-2: Security Requirements for Cryptographic Modules", December 2002, http://csrc.nist.gov/publications/fips/fips140-2/fips1402.pdf

[MS-ADA3] Microsoft Corporation, "Active Directory Schema Attributes N-Z", June 2007.

[MS-GLOS] Microsoft Corporation, "Windows Protocols Master Glossary", March 2007.

[MS-LSAD] Microsoft Corporation, "Local Security Authority (Domain Policy) Remote Protocol Specification", June 2007.

[MS-PAC] Microsoft Corporation, "Privilege Attribute Certificate Data Structure", January 2007.

[MS-PKCA] Microsoft Corporation, "Public Key Cryptography for Initial Authentication (PKINIT) in Kerberos Protocol Specification", January 2007.

[MS-RPCE] Microsoft Corporation, "Remote Procedure Call Protocol Extensions", January 2007.

[MS-SFU] Microsoft Corporation, "Kerberos Protocol Extensions: Service for User and Constrained Delegation Protocol Specification", January 2007.

[MS-SNTP] Microsoft Corporation, "Network Time Protocol (NTP) Authentication Extensions", March 2007.

[Referrals] Raeburn, K., Zhu, L., and Jaganathan, K., "Generating KDC Referrals to Locate Kerberos Realms", June 2006, http://tools.ietf.org/html/draft-ietf-krb-wg-kerberos-referrals-08

[RFC1510] Kohl, J., Neuman, C., "The Kerberos Network Authentication Service (V5)", RFC 1510, September 1993, http://www.ietf.org/rfc/rfc1510.txt

[RFC1964] Linn, J., "The Kerberos Version 5 GSS-API Mechanism", RFC 1964, June 1996, http://www.ietf.org/rfc/rfc1964.txt

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997, http://www.ietf.org/rfc/rfc2119.txt

[RFC2743] Linn, J., "Generic Security Service Application Program Interface Version 2, Update 1", RFC 2743, January 2000, http://www.ietf.org/rfc/rfc2743.txt

[RFC2279] Yergeau, F., "UTF-8, A Transformation Format of ISO10646", RFC 2279, January 1998, http://www.ietf.org/rfc/rfc2279.txt

[RFC3961] Raeburn, K., "Encryption and Checksum Specifications for Kerberos 5", RFC 3961, February 2005, http://www.ietf.org/rfc/rfc3961.txt

[RFC3962] Raeburn, K., "Advanced Encryption Standard (AES) Encryption for Kerberos 5", RFC 3962, February 2005, http://www.ietf.org/rfc/rfc3962.txt

[RFC4120] Neuman, C., Yu, T., Hartman, S., and Raeburn, K., "The Kerberos Network Authentication Service (V5)", RFC 4120, July 2005, http://www.ietf.org/rfc/rfc4120.txt

[RFC4121] Zhu, L., Jaganathan, K., and Hartman, S., "The Kerberos Version 5 Generic Security Service Application Program Interface (GSS-API) Mechanism: Version 2", RFC 4121, July 2005, http://www.ietf.org/rfc/rfc4121.txt

[RFC4556] Zhu, L., and Tung, B., "Public Key Cryptography for Initial Authentication in Kerberos", RFC 4556, June 2006 http://www.ietf.org/rfc/rfc4556.txt

[RFC4757] Jaganathan, K., Zhu, L., and Brezak, J., "The RC4-HMAC Kerberos Encryption Types Used by Microsoft Windows", RFC 4757, December 2006, http://www.ietf.org/rfc/rfc4757.txt

[X680] ITU-T, "Abstract Syntax Notation One (ASN.1): Specification of Basic Notation", Recommendation X.680, July 2002, http://www.itu.int/rec/T-REC-X.680/en

Note There is a charge to download the specification.

[X690] ITU-T, "Information Technology - ASN.1 Encoding Rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)", Recommendation X.690, July 2002, http://www.itu.int/rec/T-REC-X.690/en

Note There is a charge to download the specification.

1.2.2 Informative References

[ADDLG] Microsoft Corporation, "Security Briefs: Credentials and Delegation", September 2005, http://msdn.microsoft.com/msdnmag/issues/05/09/SecurityBriefs/default.aspx

[ADFOLDING] Microsoft Corporation, "CompareString Function", http://msdn2.microsoft.com/en-us/library/ms647476.aspx

[DIALOGUE] Bryant, B. and Ts'o, T., "Designing an Authentication System: A Dialogue in Four Scenes", February 1997, http://web.mit.edu/kerberos/www/dialogue.html

[KAUFMAN] Kaufman, C., Perlman, R., and M. Speciner, "Network Security: Private Communication in a Public World, Second Edition", Prentice Hall, 2002, ISBN: 0130460192.

[MS-ADTS] Microsoft Corporation, "Active Directory Technical Specification", June 2007.

[MS-SECO] Microsoft Corporation, "Windows Security Overview", January 2007.

[MS-SPNG] Microsoft Corporation, "Simple and Protected Generic Security Service Application Program Interface Negotiation Mechanism (SPNEGO) Protocol Extensions", January 2007.

[RFC2222] Myers, J., "Simple Authentication and Security Layer (SASL)", RFC 2222, October 1997, http://www.ietf.org/rfc/rfc2222.txt

[SPNATT] Microsoft Corporation, "<servicePrincipalName>", http://msdn2.microsoft.com/en-us/library/aa347698.aspx

[SSPI] Microsoft Corporation, "SSPI", http://msdn2.microsoft.com/en-us/library/aa380493.aspx

[UUKA-GSSAPI] Swift, M., Brezak, J., and Moore, P., "User to User Kerberos Authentication using GSS-API", October 2001, http://www.watersprings.org/pub/id/draft-swift-win2k-krb-user2user-03.txt

1.3 Protocol Overview

KILE is a security protocol that authenticates entities on a network and provides additional **services** after the parties are authenticated with each other. KILE specifies extensions to the Kerberos V5 protocol.

1.3.1 Security Background

Because KILE is a security protocol, the normative references (section 1.2.1) and this specification use terms that are commonly used in the security field. In this specification, every effort was made to use terms (such as kerberos principal, **key**, and service) in the same way that they are used in [RFC4120] section 1.7.

Any implementer who wants to understand the variations between KILE and Kerberos V5, or among all the Kerberos implementations, should have a working knowledge of the Kerberos protocol. Several informative references (section 1.2.2), specifically [DIALOGUE] and [KAUFMAN], provide an excellent high-level understanding of the Kerberos protocol and message flow. [KAUFMAN] also provides an excellent survey of other security protocols and concepts, and helps explain the terminology that is used in this document.

Finally, there are details in [RFC4120] and [RFC4121], and the predecessor documents [RFC1964], [RFC2743], and [RFC1510], that are not always immediately apparent. Careful study must be made, particularly of how **Generic Security Services (GSS)** [RFC2743] and the Kerberos implementation of GSS [RFC4121] tie together.

1.3.2 Kerberos Network Authentication Service (V5) Synopsis

The Kerberos V5 protocol provides a mechanism for mutual authentication between a client and a server before application data is transmitted between them. Kerberos V5 is composed of three exchanges described in detail in [RFC4120] sections 1.1 and 3.

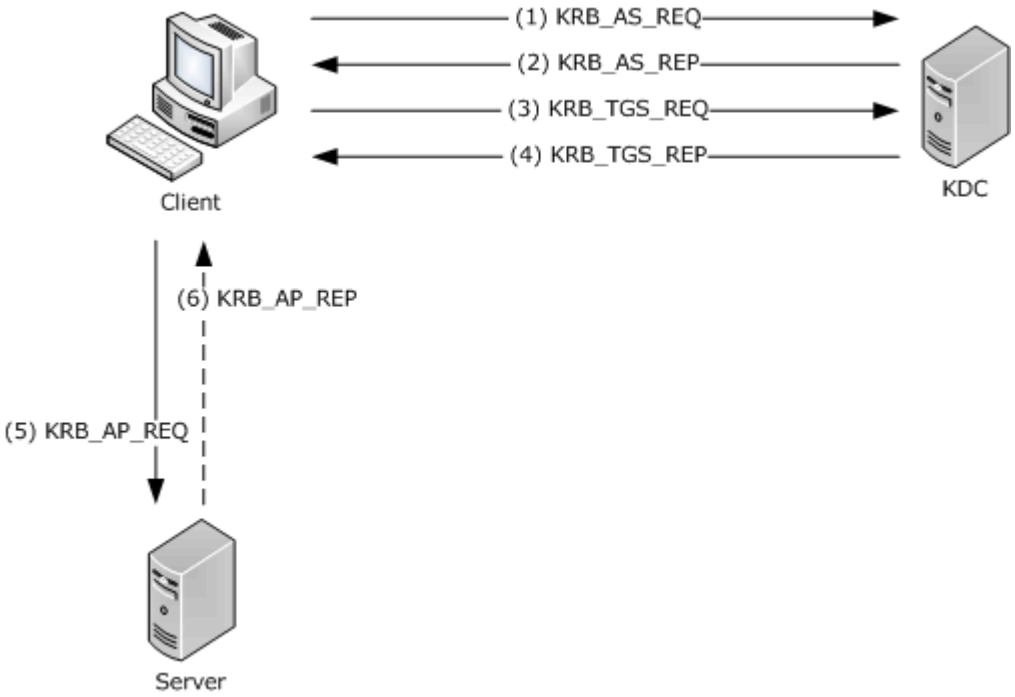

Figure 1: Kerberos V5 Exchanges

Note The terms client, server and **Key Distribution Center (KDC)**, as used in this section, refer to Kerberos V5 implementations of each entity. Unless explicitly noted, use of these terms in the remainder of this specification refers to KILE implementations of each entity.

The Authentication Service (AS) Exchange ([RFC4120] section 3.1).

- Kerberos authentication service request (KRB_AS_REQ) ([RFC4120] section 5.4.1): Using this message, the client authenticates to the KDC by persistent key (e.g., private key, symmetric key, or password-derived key). The client sends a request to the KDC for a **ticket-granting ticket (TGT)** ([RFC4120] section 5.3).
- Kerberos authentication service response (KRB_AS_REP) ([RFC4120] section 5.4.2): The KDC returns a TGT and a **session key** the client can use to encrypt and authenticate communication with the KDC for **ticket-granting service (TGS)** requests, without reusing the persistent key.

The Ticket-Granting Service (TGS) Exchange ([RFC4120] section 3.3).

- Kerberos ticket-granting service request (KRB_TGS_REQ) ([RFC4120] section 5.4.1): The client sends a request to the KDC for a ticket ([RFC4120] section 5.3) for the server. The client presents the TGT ([RFC4120] section 5.3), an **authenticator** ([RFC4120] section 5.5.1), and the **Service Principal Name (SPN)**.
- Kerberos ticket-granting service response (KRB_TGS_REP) ([RFC4120] section 5.4.2): The KDC validates the TGT ([RFC4120] section 5.3) and the authenticator ([RFC4120] section 5.5.1). If these are valid, the KDC returns a service ticket ([RFC4120] section 5.3) and session key the client can use to encrypt communication with the server.

The Client/Server Authentication Protocol (AP) Exchange ([RFC4120] section 3.2).

- Kerberos application server request (KRB_AP_REQ) ([RFC4120] section 5.5.1): The client requests access to the server. The client presents the ticket ([RFC4120] section 5.3) and a new authenticator ([RFC4120] section 5.5.1). The server will decrypt the ticket, validate the authenticator, and can use any **authorization data** ([RFC4120] section 5.2.6) contained in the ticket for access control.
- Kerberos application server response (KRB_AP_REP) ([RFC4120] section 5.5.2): Optionally, the client might request that the server verify its own identity. If mutual authentication is requested, the server returns the client's timestamp from the authenticator encrypted with the session key.

The **AS exchange** and TGS exchange are transported by Kerberos implementations. The AP exchange is passive and relies on an upper-layer application protocol to carry the **AP exchange** messages. Applications that use AP exchange messages directly are typically called "kerberized" applications. Most applications use the Generic Security Service Application Program Interface (GSS-API) and may even be wrapped by higher-level abstractions such as Simple Authentication and Security Layer (SASL) [RFC2222], which allows for "kerberized" connections to mail servers.

1.3.3 KILE Synopsis

By extending the authorization data ([RFC4120] section 5.2.6), KILE provides the server with additional information such as:

- Group membership
- Interactive logon information
- Integrity levels

By extending the KDC's account database, KILE provides control at the principal level for things such as constrained delegation and Data Encryption Standard (DES) usage.

How authorization is accomplished using **Privilege Attribute Certificate (PAC)** [MS-PAC] data is described in [MS-SECO] section 3.2.2.2.

1.4 Relationship to Other Protocols

KILE is only one part of the Windows implementation of Kerberos. KILE specifies extensions to Kerberos V5 [RFC4120].

Kerberos V5 AS and TGS exchanges rely on either the User Datagram Protocol (UDP) or the Transmission Control Protocol (TCP) ([RFC4120] section 7.2.1) as a transport. KILE relies on a working Domain Name System (DNS) infrastructure.

Kerberos V5 AP Exchange messages are only carried in other application protocols and never exist by themselves on the network. Almost any application can (theoretically) use Kerberos V5 authentication; applications that already adopt a GSS-style approach to security are most applicable. <1>

Other non-RFC standard specifications relevant to the implementation of Kerberos are:

- Authentication Protocol Domain Support Specification [MS-APDS]
- Privilege Attribute Certificate Data Structure [MS-PAC]
- Public Key Cryptography for Initial Authentication (PKINIT) in Kerberos Protocol Specification [MS-PKCA]

- Kerberos Protocol Extensions: Service for User and Constrained Delegation Protocol Specification [MS-SFU]
- User to User Kerberos Authentication using GSS-API [UUKA-GSSAPI]

1.5 Prerequisites/Preconditions

The Kerberos V5 protocol assumes the following:

- The clocks of the participants (clients, servers, and KDCs) must be synchronized within a reasonable window of time. The skew window is five minutes for most implementations of the Kerberos V5 protocol.<2>
- The KDC shares a secret key with the client and a separate secret key with the server. The provisioning of these secret keys is done out of band and is not part of KILE. Kerberos V5 implementations have a directory or database that contains at least the list of accounts and the associated secret keys. <3>
- A source of cryptographically useful random numbers is available for generating keys and other cryptographically sensitive information.

General Kerberos V5 protocol assumptions are as specified in [RFC4120] section 1.6.

1.6 Applicability Statement

The Kerberos V5 protocol provides suitable authentication for clients and servers on a network that receives some level of management. The Kerberos V5 protocol is not applicable for stand-alone machines or among machines that do not have a common management infrastructure (for example, between clients and Web servers on the Internet).

KILE is applicable to any application protocol that also requires integrated authorization and group management. These extensions are also applicable to any other use for which the Kerberos V5 protocol alone is suitable.

1.7 Versioning and Capability Negotiation

The Kerberos V5 protocol contains a version number field in many messages. However, using the version number field is an unworkable method for versioning. The wire format of the Kerberos version 4 protocol is incompatible with version 5 so that the version number is not useful. In the Kerberos V5 protocol, the value of the version number field is a fixed value of 5 ([RFC4120] section 5.2.4).

1.7.1 Pre-Authentication

The Kerberos V5 protocol supports **pre-authentication**, which takes place during the AS exchange and occurs when the client first authenticates to the KDC. A client pre-authenticates if it supplies additional information that proves it knows the key it shares with the KDC before the TGT is issued. See Pre-authentication Data (section 3.1.5.1) for a complete specification of these types supported by KILE.

1.7.2 Encryption Types

The Kerberos V5 protocol supports multiple encryption types, which are the actual algorithms for encrypting the **tickets** or other data. The Kerberos V5 protocol negotiates which encryption type to use for a particular connection ([RFC4120] section 3.1.3). See Encryption Types (section 3.1.5.2) for a complete specification of these types supported by KILE.

1.8 Vendor-Extensible Fields

The Kerberos V5 protocol includes several areas for vendor extension.

KILE does not provide vendor extensibility beyond what is specified in [RFC4120].

1.9 Standards Assignments

Assignment of Kerberos V5 IANA numbers is as specified in [RFC4120] section 9. UDP port 88 and TCP port 88 are used when communication between the client and the KDC occurs.

2 Messages

2.1 Transport

The Kerberos V5 protocol uses UDP and TCP for transport ([RFC4120] section 7.2). KILE SHOULD use UDP by default; however, if the message size exceeds a specific configurable value (message size threshold), TCP SHOULD be used <4>. The threshold applies to **AS** and TGS messages. They do not apply to AP messages because the transport is controlled by the application protocol.

KILE MUST have a working DNS infrastructure. KILE SHOULD NOT use the Internet Protocol (IP) addresses of the KDCs <5>.

2.2 Message Syntax

KILE does not alter the syntax of any Kerberos V5 messages ([RFC4120] sections 5.4 through 5.9). KILE extensions provide platform-specific data to support encoding authorization data ([MS-PAC] section 2) in the authorization data field ([RFC4120] sections 5.2.6 and 5.2.7) of the ticket.

The authorization data, which MUST be encoded as a PAC, MUST be marked as AD-IF-RELEVANT, which means that it SHOULD be ignored by implementations that do not understand the format.

Kerberos V5 messages are defined using Abstract Syntax Notation One (ASN.1), as specified in [X680], and encoded using Distinguished Encoding Rules (DER), as specified in [X690] section 10.

2.2.1 KERB-PA-PAC-REQUEST

This structure is a PA-DATA type that is defined to explicitly request a PAC in the ticket. Its structure is defined using ASN.1 notation and the syntax is as follows:

```
KERB-PA-PAC-REQUEST ::= SEQUENCE {
include-pac[0] BOOLEAN --If TRUE, and no pac present, include PAC.
                       --If FALSE, and PAC present, remove PAC
}
```

2.2.2 LSAP_TOKEN_INFO_INTEGRITY

The **LSAP_TOKEN_INFO_INTEGRITY** structure specifies the **integrity level** information for the client <6>.

```
typedef struct _LSAP_TOKEN_INFO_INTEGRITY {
  unsigned long Flags;
  unsigned long TokenIL;
  unsigned char MachineID[32];
} LSAP_TOKEN_INFO_INTEGRITY,
 *PLSAP_TOKEN_INFO_INTEGRITY;
```

Flags: A 32-bit unsigned integer indicating the token information type. This value MUST be one of the following:

Value	Meaning
0x00000000	Full token.
0x00000001	User Account Control (UAC) restricted token.

TokenIL: A 32-bit unsigned integer indicating the token's integrity level. This value MUST be one of the following:

Value	Meaning
0x00000000	Untrusted.
0x00001000	Low.
0x00002000	Medium.
0x00003000	High.
0x00004000	System.
0x00005000	Protected process.

MachineID: A 32-byte binary random string created at computer startup used as an ID to detect loopback authentication.

2.2.3 KERB-AD-RESTRICTION-ENTRY

The KERB-AD-RESTRICTION-ENTRY structure specifies additional restrictions for the client.<7> Its structure is defined using ASN.1 notation and the syntax is as follows:

```
KERB-AD-RESTRICTION-ENTRY ::= SEQUENCE {
restriction-type            [0] Int32,
restriction                 [1] OCTET STRING
}
```

Restriction-Type: MUST be set to 0x00000000.

Restriction: An **LSAP_TOKEN_INFO_INTEGRITY** structure that contains the integrity information for the client.

3 Protocol Details

This section specifies details of KILE, including abstract data models and message processing rules, as follows:

- Common Details (section 3.1) specifies extensions to common elements.
- Client Details (section 3.2) specifies extensions specific to the client during the AS, TGS and AP exchanges.
- KDC Details (section 3.3) specifies extensions specific to the KDC processing of AS and TGS requests.
- Application Server Details (section 3.4) specifies extensions to the server processing of the AP requests.

3.1 Common Details

3.1.1 Abstract Data Model

Kerberos V5 specifies the abstract data model for common elements.

KILE specifies the following extensions to common elements:

- Replay Cache
- Cryptographic Material
- Ticket Cache
- Machine ID

3.1.1.1 Replay Cache

Kerberos V5 specifies that servers MUST utilize a replay cache unless the application server provides replay protection ([RFC4120] section 3.2.3).

KILE MUST implement a replay cache regardless of the application server replay functionality.

3.1.1.2 Cryptographic Material

Kerberos V5 establishes a secret key that is shared by a principal and the KDC and a session key that forms the basis for privacy or integrity in the communication channel between client and server.

Using KILE, application clients (for example, CIFS/SMB clients) MAY use the negotiated key directly. When an application client uses the session key, the application protocol MUST document the explicit use of the key in its protocol specification. The key MAY be exported as an attribute of the completed security context in the **SSPI** API.

3.1.1.3 Ticket Cache

Kerberos V5 specifies that servers MAY cache TGTs ([RFC4120] section 3.3.1).

KILE MAY implement a ticket cache that preserves service tickets and TGTs.<8>

3.1.1.4 Machine ID

KILE MUST implemement a machine ID for use in loopback detection for integrity levels.

3.1.2 Timers

There are no common timers.

3.1.3 Initialization

The random number generator for keys and nonces is initialized by other components but complies with [FIPS140] section 4.7.1.

3.1.4 Higher-Layer Triggered Events

There are no common higher-layer triggered events.

3.1.5 Message Processing Events and Sequencing Rules

The following sections detail variations in tickets and naming that are common to all parts of the Kerberos protocol.

3.1.5.1 Pre-authentication Data

Pre-authentication ([RFC4120] sections 3.1.1, 5.4.1, and 5.2.7) is an extensibility point for the Kerberos V5 protocol. Pre-authentication is performed by supplying one or more pre-authentication messages in the PA-data field of the AS-REQ message.

KILE supports the following pre-authentication types ([RFC4120] section 7.5.2):

- PA-TGS-REQ [1]
- PA-ENC-TIMESTAMP [2]
- PA-ETYPE-INFO [11]
- PA-PK-AS-REP_OLD [15]
- PA-PAC-REQUEST [128]

Unknown pre-authentication types MUST be ignored by KDCs. If the KDC cannot find a satisfactory pre-authentication message in the AS exchange, an error MUST be returned to the client. The exact error depends on what pre-authentication types were supplied.

When clients perform a password-based initial authentication, they MUST supply the PA-ENC-TIMESTAMP pre-authentication type when they construct the initial AS request. They MAY <9> request, via the PA-PAC-REQUEST pre-authentication type, that a privilege attribute certificate (PAC) be included in issued tickets. KILE supports PKINIT [RFC4556] and the Public Key Cryptography for Initial Authentication (PKINIT) in Kerberos Protocol Specification [MS-PKCA].

3.1.5.2 Encryption Types

KILE MUST support the Advanced Encryption Standard (AES) encryption types and MAY <10> support the other following encryption types, which are listed in order of relative strength:

- AES256-CTS-HMAC-SHA1-96 [17] ([RFC3962] section 7) <11>

- AES128-CTS-HMAC-SHA1-96 [18] ([RFC3962] section 7) <12>
- RC4-HMAC [23] [RFC4757]
- RC4-HMAC-EXP [24] [RFC4757]
- DES-CBC-MD5 [3] [RFC3961]
- DES-CBC-CRC [1] [RFC3961]

Kerberos V5 encryption type assigned numbers are specified in [RFC3961] section 8, [RFC4757] section 5 and [RFC3962] section 7.

3.1.5.3 Ticket Flag Details

The Kerberos V5 protocol specifies a number of options and behaviors with regard to the flags ([RFC4120] section 2) that are encoded in a ticket.

KILE implements the following ticket flags:

- The INITIAL and PRE-AUTHENT flags ([RFC4120] section 2.1): By default, KDCs require pre-authentication when they issue tickets. Clients SHOULD pre-authenticate. KDCs MUST enforce pre-authentication therefore, unless the account has been explicitly set to not require Kerberos pre-authentication, the ticket will have the PRE-AUTHENT flag set.
- The HW-AUTHENT flag ([RFC4120] section 2.1): This flag was originally intended to indicate that hardware-supported authentication was used during pre-authentication. This flag is no longer recommended in the Kerberos V5 protocol. KDCs MUST NOT issue a ticket with this flag set. KDCs SHOULD NOT preserve this flag if it is set by another KDC.
- The RENEWABLE flag ([RFC4120] section 2.3): Renewable tickets SHOULD be supported in KILE.
- The POSTDATED/MAY-POSTDATE flag ([RFC4120] section 2.4): Postdated tickets SHOULD NOT be supported in KILE.
- The PROXY/PROXIABLE flag ([RFC4120] section 2.5): Proxiable tickets SHOULD NOT be supported in KILE.
- The FORWARDABLE/FORWARDED flag ([RFC4120] section 2.6): Forwarded tickets SHOULD be supported in KILE.
- The TRANSITED-POLICY-CHECKED flag ([RFC4120] section 2.7): KILE MUST NOT check for transited domains on servers or a KDC. Application servers MUST ignore the TRANSITED-POLICY-CHECKED setting.
- The OK-AS-DELEGATE flag ([RFC4120] section 2.8): The KDC MUST set the OK-AS-DELEGATE flag if the service account is trusted for delegation (section 3.3.1.1). For more information, see [ADDLG].

3.1.5.4 Other Elements and Options

The Kerberos V5 protocol defines pre-authentication data ([RFC4120] section 5.2.7) and optional authorization data elements ([RFC4120] section 5.2.6) for use in tickets and authenticators.

KILE MUST support the PA-ENC-TIMESTAMP pre-authentication data type ([RFC4120] section 5.2.7.2) and the AD-IF-RELEVANT element ([RFC4120] section 5.2.6.1).

KILE SHOULD NOT support the following elements:

- The AD-KDC-ISSUED element ([RFC4120] section 5.2.6.2).
- The AD-AND-OR element ([RFC4120] section 5.2.6.3).
- The AD-MANDATORY-FOR-KDC element ([RFC4120] section 5.2.6.4).

KILE SHOULD NOT fail on unknown authorization data ([RFC4120] section 1.5.1). The server SHOULD NOT generate an error; instead, it SHOULD ignore the unknown data and proceed to authenticate the client.

KILE MUST support the KRB_ERR_RESPONSE_TOO_BIG error message ([RFC4120] section 7.2.1).

3.1.5.5 Addressing

KILE MUST support IPv6 addresses ([RFC4120] section 7.1) <13>.

KILE MUST NOT support directional addresses ([RFC4120] section 7.1). If the directional addresses are present, they MUST be ignored.

3.1.5.6 Internationalization and Case Sensitivity

The Kerberos V5 protocol specifies rules for encoding and processing names, both for character set and case ([RFC4120] section 6).

User names and **domain** name comparisons MUST NOT be case sensitive in the KILE. KILE MUST use UTF-8 encoding of these names [RFC2279]. Normalization MUST NOT be performed and surrogates MUST NOT be supported.<14>

3.1.5.7 Key Version Numbers

The Kerberos V5 protocol specifies key version numbers ([RFC4120] section 5.2.9). Key version numbers are used in the Kerberos V5 protocol to distinguish between different keys in the same domain.

When branch office domains are present, the key version numbers are segmented for each branch.<15> This segmentation allows the domain controller to distinguish between keys that are issued by different branches.

The key version number consists of 32 bits. The first 16 bits identify the branch and the remaining 16 bits identify specific keys in a branch.

3.1.5.8 Referrals

The Kerberos V5 protocol specifies cross-**realm** behavior and the nature of referrals ([RFC4120] section 1.2).

KILE MUST support cross-realm referrals ([RFC4120] sections 1.2 and 3.3.1) and extended referrals [Referrals].

3.1.5.9 PAC Generation

The PAC [MS-PAC] MUST be generated by the KDC under one of the following conditions:

- During an Authentication Service (AS) request that has been validated with pre-authentication.

- During a TGS request when the TGT for the client in the request does not contain a PAC and the ticket to be returned is a cross-realm referral TGT ([RFC4120] section 1.2).
- During a TGS request when the client has domain local groups.

The KDC MUST collect the user's initial set of group information and add it to the PAC in the TGT.

The PAC MUST be subsequently updated when the client requests a **service ticket** to contain additional domain local groups that are specific to the server's domain.

By default, the KDC MUST generate a PAC. However, a client MAY <16> explicitly request that a PAC be included or excluded through the use of a KERB-PA-PAC-REQUEST PA-DATA type (2.2.1).

3.1.6 Timer Events

No timer events exist other than those specified in [RFC4120].

3.1.7 Other Local Events

No local events other than those specified in [RFC4120] are needed.

3.1.8 Implementing Public Keys

The use of public keys in KILE is specified in [MS-PKCA].

3.2 Client Details

3.2.1 Abstract Data Model

The Kerberos V5 protocol specifies the client abstract data model.

3.2.1.1 Application Parameters

The following parameters are logically available for the application to set. These logical parameters can influence various protocol-defined flags.<17>

Note: The variables that are defined below are logical, abstract parameters that an implementation will have to maintain and expose to provide the correct level of service.

- Delegate: The Delegate flag indicates that the client should set the FORWARDABLE option in the TGS request. When the client receives a forwardable ticket, it puts the ticket in a KRB_CRED structure ([RFC4120] section 3.6). The client does not forward the ticket unless the TGT is marked OK-AS-DELEGATE ([RFC4120] section 2.8) and the service is a suitable recipient of delegation (section 3.1.5.3).
- Mutual Authentication: The Mutual Authentication flag indicates that the client requires authentication of the server. Setting this flag results in setting the MUTUAL-REQUIRED flag in the KRB_AP_REQ message ([RFC4120] section 3.2.2 and section 3.2.4). Even with this flag, mutual authentication cannot be assured until the first message is passed by the application protocol and the message is signed or encrypted.
- DCE Style: The DCE Style flag indicates that the caller wants three-leg, DCE-style authentication ([MS-RPCE] and [C706]). To prevent man-in-the-middle (MITM) attacks, an application protocol must sign or encrypt its messages after authentication. This flag can also establish mutual authentication, which is not truly established until both parties are shown to use the session key on "fresh" messages. DCE originally opted to have one more client-to-server message that would

demonstrate this key usage, at the expense of some efficiency. This was termed "three-leg authentication."

This flag was added for use with MS-RPCE, which initially expected three legs of authentication. Setting this flag causes an extra AP_REP to be sent from the client back to the server after receiving the AP_REP of the server. In addition, the security tokens that are exchanged during the authentication attempt do not have GSS-API formatting applied ([RFC2743]). They are AP exchange messages with no object-identifier wrapping. DCE style is specified in section 3.4.5.1.

3.2.1.2 Security Context Parameters

After a connection is established through the AP exchange, Kerberos V5 does not and cannot directly influence the application protocol. The following parameters MUST be set when establishing the security context to support signing or encrypting messages. The higher-layer application protocol will invoke the per-message functions.

- Integrity: Indicates that the caller wants to sign messages so that they cannot be tampered with while in transit. Setting this flag results in the GSS_C_INTEG_FLAG being set in the authenticator's checksum field ([RFC1964] section 1.1.1 and [RFC4121] section 4.1.1).
- Replay Detect: Indicates that the caller wants replay detection so the application can determine when messages are replayed. Setting this flag results in the GSS_C_REPLAY_FLAG being set in the authenticator's checksum field ([RFC1964] section 1.1.1 and [RFC4121] section 4.1.1).
- Sequence Detect: Indicates that the caller wants sequence detection so that messages cannot be received out of order. Setting this flag results in the GSS_C_SEQUENCE_FLAG being set in the authenticator's checksum field ([RFC1964] section 1.1.1 and [RFC4121] section 4.1.1).
- Confidentiality: Indicates that the caller wants to encrypt messages so that they cannot be read while in transit. Setting this flag results in the GSS_C_CONF_FLAG being set in the authenticator's checksum ([RFC1964] section 1.1.1 and [RFC4121] section 4.1.1).
- Mutual Authentication: Indicates that the caller wants to authenticate the identity of the client to the server and the server to the client. This flag corresponds to the Mutual Authentication option. For more information, see [RFC2743] section 1.2.1.2; the Kerberos Authentication Protocol option MUTUAL-REQUIRED ([RFC4120] section 3.2.4 and section 5.5.1); and the GSS_C_MUTUAL_FLAG ([RFC1964] section 1.1.1).
- Extended Error: Indicates that the caller wants to receive additional error handling, including possibly retries, with the context of the Generic Security Services (GSS) exchange in progress. Setting this flag results in the GSS_C_EXTENDED_ERROR_FLAG being set in the authenticator's checksum field ([RFC1964] section 1.1.1, [RFC4121] section 4.1.1, and [RFC4757] section 7.1).
- Identify: Indicates that the caller wants the server to know the identity of the caller but not be allowed to impersonate the caller to resources on that system.

  Corresponds to the Windows security model. Indicates that the GSS_C_IDENTIFY_FLAG was set in the GSS_Init_sec_context call ([RFC4757] section 7.1), and results in the GSS_C_IDENTIFY_FLAG set in the authenticator's checksum field ([RFC1964] section 1.1.1, [RFC4121] section 4.1.1, and [RFC4757] section 7.1).
- Datagram Style: Initializes the security context [RFC1964], but does not transmit the authentication message. Indicates that the caller wants to use Datagram semantics (see Datagram-Style (section 3.4.5.2)).

3.2.2 Timers

When the client sends an AS-REQ or TGS-REQ to the KDC, it uses a timer to determine when to retry. The operation of this timer, along with its default values, is as specified in section 3.2.6.

3.2.3 Initialization

Before the client can send an AS or TGS message, it MUST discover which KDC to send the AS or TGS message. Clients SHOULD use **SRV record** discovery ([RFC4120] section 7.2.3.2)<18> by default. Clients MAY<19> use a list of KDCs for a specified realm.

If the client has a ticket cache, the ticket cache MUST be initialized to an empty state.

3.2.4 Higher-Layer Triggered Events

3.2.4.1 Initial Logon

Initial logon is the process by which a user first authenticates to the KDC. The client engages in an AS exchange (see section 1.3.2) with the KDC, using domain password or smartcard authentication and receiving a TGT and session key. The TGT and session key are then used in subsequent protocol exchanges with the KDC in requesting service tickets.

The client SHOULD request a service ticket to its own workstation during initial logon from the KDC because the service ticket contains information about the logged on user contained in the user's PAC within the service ticket. The client MAY<20> use the information in that PAC for access control purposes.

For initial authentication using PKINIT ([RFC4556] and [MS-PKCA]), the KDC will also return a set of credentials for use with other authentication protocols ([MS-PAC]).

Standard Kerberos requires that the user principal name (UPN) refer to a valid domain the KDC defines (e.g., user@windows.example.com). KILE permits the UPN to refer to an apparent domain (e.g., user@example.com).<21>

3.2.4.2 Authentication to Services

When the initial authentication is complete and the TGT is obtained, the user typically wants to use a network resource. . For a Kerberos-aware application, the Kerberos client initiates a **TGS exchange** requesting a service ticket to the named service, for example "host/hostname.domain.name".

The Kerberos client then initiates an AP exchange which MAY be encoded in a GSS–API style wrapper, if the Kerberos-aware application requests it.

KILE provides no support for direct access to the Kerberos **KRB_SAFE** or **KRB_PRIV** messages.

The client application then takes the AP message and supplies it, in band with the application protocol, to the server. The Kerberos server processes the message as specified in [RFC4120] and completes the connection. The AP exchange is covered further in section 3.3.

3.2.5 Message Processing Events and Sequencing Rules

3.2.5.1 Request Flags Details

Kerberos V5 specifies Kerberos ticket-issuing behavior modification defined by a set of options that are passed to the KDC during the AS exchange or TGS exchange.

Clients always set the canonicalize flag ([RFC4120] section 5.4.1).

3.2.5.2 AS Exchange

The Kerberos V5 protocol specifies the AS exchange ([RFC4120] section 3.1). KILE also supports extensions to the AS exchange as specified in [Referrals], [RFC4556], and [MS-PKCA].

The client will always include a PAC request PA-data type when generating an AS-REQ message. The PAC is specified in [MS-PAC].

3.2.5.3 AP Exchange

The client MUST send integrity level information in the AP request as an authorization data field ([RFC4120] section 5.2.6) of type KERB_AUTH_DATA_TOKEN_RESTRICTIONS (141), containing the KERB-AD-RESTRICTION-ENTRY structure (section 2.2.3).<22>

3.2.6 Timer Events

The Kerberos V5 protocol requires the client to contact the KDC and recognizes that a specific KDC could be offline or unavailable to service the request. The actual behavior is not specified in [RFC4120]; these behavior details are determined by the implementation. Detection of a KDC's failure to reply requires a timer.<23>

3.2.7 Other Local Events

No local events other than those specified in [RFC4120] are required.

3.3 KDC Details

3.3.1 Abstract Data Model

In addition to the abstract data model and default values specified in Kerberos V5, KILE uses the following configuration values:

- Minimum lifetime: 0 minute
- Maximum ticket lifetime:<24> 10 hours.

KILE introduces a KDC configuration setting:<25>

- AuthenticationOptions: This is a flag field. Only AUTH_REQ_VALIDATE_CLIENT is supported and SHOULD be set by default

3.3.1.1 Account Database Extensions

The Kerberos V5 protocol specifies that KDCs MUST maintain a database of principals and their secret keys.

To support all functionality of KILE, the account database MUST be extended to support the following additional information for each principal:

- Authorization data not required: When this flag is set on the principal, the KDC MUST NOT include a PAC in the service ticket.<26>
- Delegation not allowed: When this flag is set on the principal, the KDC MUST NOT set the PROXIABLE or FORWARDABLE ticket flags ([RFC4120] sections 2.5 and 2.6).<27>

- Group membership: The account database MUST be extended to support groups to support PAC generation. For more information, see section 3.3.5.3.2 and section 3.3.5.4.2.
- Pre-Authentication not required: When this flag is set on the principal, the KDC MUST issue a TGT without valid pre-authentication data ([RFC4120] section 7.5.2) provided.<28>
- Realms or domains: The account database MUST have a larger database of accounts and their realms or domains to support referrals. For more information, see section 3.3.5.3.1.
- Services allowed to send forwarded tickets to: The account database MUST be extended to support the list of services to which a service can forward tickets to support constrained delegation. For more information, see section 3.3.5.4.4.
- Trusted to authentication for delegation: When this flag is set on the principal and the service obtains an S4USelf [MS-SFU] service ticket, the KDC MUST set the FORWARDABLE ticket flag ([RFC4120] section 2.6). When this flag is not set, the KDC MUST NOT set the FORWARDABLE ticket flag ([RFC4120] section 2.6) in the S4USelf service ticket.<29>
- Trusted for delegation: When this flag is set on the principal, the KDC MUST set the OK-AS-DELEGATE ticket flag ([RFC4120] section 2.8).<30>
- Use DES only: When this flag is set on the principal, only the des-cbc-md5 and/or des-cbc-crc keys [RFC3961] are used in the Kerberos exchanges for this account.<31>

3.3.2 Timers

There are no KDC timers.

3.3.3 Initialization

Kerberos V5 specifies that all KDCs in a domain MUST have the same key<32> and the name of the service for the TGS is "krbtgt/domain-name" SPN ([RFC4120] section 6.2).

If the KDC has a ticket replay cache, it MUST be reset when the KDC starts up.

If the KDC has a ticket cache, the ticket cache MUST be initialized to an empty state.

3.3.4 Higher-Layer Triggered Events

There are no KDC higher-layer trigger events.

3.3.5 Message Processing Events and Sequencing Rules

3.3.5.1 Request Flag Details

Kerberos V5 specifies Kerberos ticket-issuing behavior modification defined by a set of options that are passed to the KDC during the AS or TGS exchange.

KDCs SHOULD<33> ignore the canonicalize flag except for referrals [Referrals].

Canonicalization was designed to allow aliasing for principals. This allowed the client to request a ticket to "cifs/hostname" and the KDC to issue a ticket to "host/hostname" which allowed for exposing the "true" name of the principal. This behavior resulted in inefficiencies and confusion for several reasons:

- The client ticket cache became unusable because all the tickets were named "host/hostname" and a cache lookup for "cifs/hostname" never succeeded.
- Third-party implementations of the Kerberos-aware applications that used the Kerberos protocol expected the name in the ticket to match the requested name and ran into problems when they did not. This confusion was mitigated by disabling strict name checking in the third-party implementations when they interoperate with older versions of KILE.

3.3.5.2 Encryption Supported

The KDC MUST<34> return in the AS-REP and TGS-REP messages PA-DATA with padata-type set to PA-SUPPORTED-ENCTYPES (165), to indicate what encryption types are supported by the server or service.

The data in the padata-value field contains a 32-bit unsigned integer that contains a combination of the following flags and which specifies what encryption types are supported by the server or service.

0	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	20	1	2	3	4	5	6	7	8	9	30	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	F	E	D	C	B	A

Where the bits are defined as:

Value	Description
A	DES-CBC-CRC
B	DES-CBC-MD5
C	RC4-HMAC-EXP
D	RC4-HMAC
E	AES128-CTS-HMAC-SHA1-96
F	AES256-CTS-HMAC-SHA1-96

3.3.5.3 AS Exchange

Kerberos V5 specifies the AS exchange ([RFC4120] section 3.1). KILE also supports extensions to the AS exchange specified in [Referrals], [RFC4556] and [MS-PKCA].

3.3.5.3.1 Referrals

The KDC supports referral processing [Referrals], sending a KDC and domain to use to answer a client's request.

3.3.5.3.2 Initial Population of the PAC

During processing of the AS request, the KDC searches the AD for the user or computer account that matches the name that was sent in the AS-REQ message. The KDC queries **AD** for the set of groups of which the account is a member, both directly and transitively. The KDC then creates the PAC

structure [MS-PAC] and encodes that into the TGT using the AD-IF-RELEVANT element ([RFC4120] section 5.2.6.1).

3.3.5.4 TGS Exchange

Kerberos V5 specifies the TGS exchange ([RFC4120] section 3.3).

KILE supports the following extensions to the TGS exchange:

- AUTH_REQ_VALIDATE_CLIENT
- TGT without a PAC
- Domain Local Group Membership
- Constrained Delegation
- Cross-Domain Trust and Referrals

If the PAC contains the SID S-1-5-32-1000, the PAC MUST be used to perform an access check for the Allowed-To-Authenticate right against the AD object of the account for which the service ticket request is being made. If the access check succeeds, the service ticket MUST be issued; otherwise, the request MUST be denied.

3.3.5.4.1 AUTH_REQ_VALIDATE_CLIENT Flag

Kerberos V5 does no enforcement of revocation of accounts prior to the expiration of issued tickets. KILE introduces that enforcement by specifying that when the AUTH_REQ_VALIDATE_CERT flag is set in the account, the KDC of the same domain as the client, on processing a TGT older than 20 minutes, MUST verify that the account is still in good standing, meaning the account has not expired, been locked out, been disabled or otherwise is not allowed to log on.

If the client is not allowed to log on, the TGS request MUST fail with the errors in the following table:

Error Condition	KDC Error
The account has expired.	KDC_ERR_CLIENT_REVOKED
The account has been locked out.	KDC_ERR_CLIENT_REVOKED
The account has been disabled.	KDC_ERR_CLIENT_REVOKED
The account is not allowed to log on at present.	KDC_ERR_CLIENT_REVOKED
The account password has expired	KDC_ERR_KEY_EXPIRED
The account password must be changed now.	KDC_ERR_KEY_EXPIRED

3.3.5.4.2 TGT without a PAC

If a TGS request includes a TGT without a PAC, the KDC SHOULD add a PAC before issuing the service ticket. This occurs when the TGT was issued by a pure realm [RFC4120] that is trusted by the domain. There are two cases for which the PAC MUST be inserted:

- If the KDC is configured locally to map principals in the realm to accounts based on name [RFC4120]. In this case, the KDC MUST search the mapping for a principal with the same name.

- If there is no default mapping rule established, the KDC MUST search the AD for an account which is associated with the name in the TGT.

If a matching account is found, the KDC MUST use that account to construct a PAC and insert it into the resulting service ticket. If no account can be found, the service ticket MUST be issued without a PAC.

3.3.5.4.3 Domain Local Group Membership

Groups can be created so that they are only visible to servers in the same domain. For every service ticket that is issued during a TGS request, except for cross-realm TGTs, the KDC MUST invoke the AD to determine if any of the identities in the PAC are for domain local groups. If any are found, the PAC MUST be regenerated with the domain local groups and inserted into the ticket before it is issued to the caller.

3.3.5.4.4 Constrained Delegation

Constrained delegation [MS-SFU]<35> limits a client from being able to use a forwarded TGT to any service in the domain to only specific servers.<36>

The KDC MUST add the list of valid targets in the TGT as an additional PAC element. When issuing a service ticket, if this element is present in the TGT, the KDC MUST validate the target of the request against the services listed. If the requested target is not in the list, the request MUST fail.

3.3.5.4.5 Cross-Domain Trust and Referrals

The KDC derives its knowledge of cross-domain trusts from trusted domain objects (TDOs) in Active Directory. For more information, see [MS-ADTS].

If a cross-domain referral is determined to be necessary ([RFC4120] section 1.2 and [Referrals]), the appropriate inter-realm key MUST be retrieved from the TDO and used as specified in [RFC4120].

3.3.5.5 Naming

Kerberos V5 specifies a variety of name types ([RFC4120] section 7.5.8) for specifying the name of the server during a TGS request.

KILE SHOULD use service principal names (SPNs) to identify servers in TGS-REQs. An SPN is a single-string representation of a Kerberos principal name according to section 2.1.1 of [RFC1964] that identifies the server. The Directory Service attribute servicePrincipalName, as defined in [MS-ADA3] section 2.252, is a multi-value attribute on a user or computer object that contains a set of service principal names, with each component corresponding to a string representation of a Kerberos name that can be used to identify the server. For more information see [SPNATT].

An SPN is of the format <service class>/<host>:<port> where:

- <service class> is a string that identifies the class of the service, such as "www" for a web service or "ldap" for a directory service.
- <host> is a string that is the name of the system. This SHOULD be the fully qualified domain name (FQDN).
- Optionally, <port> is the port number for the service.

3.3.6 Timer Events

KILE introduces no timer events other than those specified in [RFC4120].

3.3.7 Other Local Events

KILE introduces no local events, other than those specified in [RFC4120].

3.4 Application Server Details

Kerberos V5 defines a protocol subordinate to some other application protocol, via GSS-API [RFC4121]. KILE extends GSS-API (see GSSWrapEx (section 3.4.5.4) and GSSUnwrapEx (section 3.4.5.5)).

The AP exchange is controlled by several logical parameters that are passed in by the higher-layer application protocol that is invoking KILE.

3.4.1 Abstract Data Model

3.4.1.1 Application Parameters

The application parameters that comprise part of the abstract data model for the Application Server are identical to those specified section 3.2.1.1.

3.4.1.2 Security Context Parameters

The security context parameters that comprise part of the abstract data model for the Application Server are identical to those specified section 3.2.1.2.

3.4.2 Timers

The AP exchange does not require specific timers.

3.4.3 Initialization

All parameters that are specified in section 3.4.1 are reset and then set according to the higher-layer protocols request.

The replay cache MUST be initialized with no entries.

3.4.4 Higher-Layer Triggered Events

The AP exchange is triggered by a higher-layer application protocol that requests security services for a connection or message exchange. The higher-layer application protocol MUST specify the name of the server to which it is attempting authentication and also MUST specify any of the parameters from section 3.4.1 that are required for Kerberos V5 [RFC4120] to perform the authentication.

Calling applications use the SSPI API family to establish the connection and specify the target. Optionally, certain higher-layer protocols, such as Simple and Protected Generic Security Service Application Program Interface Negotiation Mechanism (SPNEGO) [MS-SPNG], will also specify the parameters.

3.4.5 Message Processing Events and Sequencing Rules

Kerberos V5 specifies several additional messages ([RFC4120] sections 3.4 through 3.6) that are associated with the session after the AP exchange has completed.

KILE does not support KRB_SAFE messages ([RFC4120] section 3.4).

KILE does not support KRB_PRIV messages with a time stamp ([RFC4120] section 3.5). KILE does support KRB_PRIV messages with a sequence number ([RFC4120] section 3.5).

KILE does support KRB_CRED messages ([RFC4120] section 3.6).

3.4.5.1 Three-Leg DCE-Style Mutual Authentication

An application protocol using the Kerberos protocol must exchange application protocol messages with Kerberos signing or encryption applied in order to verify mutual authentication. DCE, in the authn_dce_secret authentication service (as specified in [C706]) mandated that mutual authentication be verified before any RPC messages were exchanged. To accommodate that requirement, the DCE Kerberos implementation issued an additional AP_REPLY message from the client to the server as part of the AP exchange subprotocol.

Kerberos V5 is not interoperable with the DCE authn_dce_secret security protocol. KILE MUST have compatible extensions for third-party extensions. KILE emulates this behavior as follows:

- The AP-REQ message MUST NOT have GSS-API wrapping. It is sent as is without encapsulating it in a header ([RFC2743] section 3.1).
- The signature message and the encryption message MUST NOT include the length of the application data; they are no longer RFC 1964–compliant [RFC1964].
- The client MUST generate an additional AP reply message exactly as the server would ([RFC4120] section 3.2.4) as the final message to send to the server. In GSS terms, the client must return success and a message to the server. It is up to the application to deliver the message to the server.
- The server MUST receive the additional AP reply message and verify that the message is constructed correctly ([RFC4120] section 3.2.5).

3.4.5.2 Datagram-Style Authentication

Datagram-style authentication is another DCE RPC-inspired variation. In summary, datagram style initializes the security context but does not transmit the authentication message. Instead, the first application data packet is signed or encrypted as decided by the higher-level application protocol and sent to the server. The server, presented with a packet for which it has no security context, sends a demand for authentication back to the client. At that point, the client sends the authentication token previously obtained from the authentication mechanism. Authentication proceeds as normal.

When authentication is complete, the server verifies or decrypts the application packet. An application protocol that uses this datagram capability MUST have the means within the application protocol to indicate the nature of the security mechanism that is used (if mechanisms other than the Kerberos V5 protocol are possible), and the nature of the protection (signature or encryption) that is applied to the application protocol message. For DCE RPC the application packet is not retransmitted. Therefore, the session key that will be used MUST be decided by the client before any communication with the server. This precludes the sub-session key option of the Kerberos V5 protocol.

3.4.5.3 Processing Authorization Data

Kerberos V5 specifies rules for processing the authorization data field in [RFC4120] section 5.2.6.

KILE MUST unpack the authorization data field ([RFC4120] section 5.2.6) and look for an AD-WIN2K-PAC structure ([RFC4120] section 7.5.4). If the structure is valid according to the PAC specification [MS-PAC], it SHOULD be used as the authorization information.

The server MUST unpack the authorization data field and look for a KERB_AUTH_DATA_TOKEN_RESTRICTIONS [141] Authorization Data Type. If the structure is a valid KERB-AD-RESTRICTION-ENTRY structure (section 2.2.3), then the server SHOULD use it as authorization information.<37>

3.4.5.4 GSS_WrapEx() Call

This call is an extension to GSS_Wrap ([RFC2743] section 2.3.3) that passes multiple buffers.

Inputs:

- context_handle CONTEXT HANDLE
- qop_req INTEGER -- 0 specifies default Quality of Protection (QOP)
- input_message ORDERED LIST of:
  - conf_req_flag BOOLEAN
  - sign BOOLEAN
  - data OCTET STRING

Outputs:

- major_status INTEGER
- minor_status INTEGER
- conf_state BOOLEAN
- output_message ORDERED LIST (in same order as input_message) of:
  - conf_state BOOLEAN
  - signed BOOLEAN
  - data OCTET STRING
- signature OCTET STRING

This call is identical to GSS_Wrap, except that it supports multiple input buffers. Input data buffers for which conf_req_flag==TRUE are encrypted in output_message. Input data buffers for which sign==TRUE are included in the message, as specified in section 3.4.5.4.1.

3.4.5.4.1 Kerberos Binding of GSS_WrapEx()

Kerberos GSS_WrapEx() depends on the encryption type of the session key for the context. The algorithms depend on which Kerberos encryption ciphers are negotiated by the Kerberos protocol.

If the session key encryption type is AES128-CTS-HMAC-SHA1-96 or AES256-CTS-HMAC-SHA1-96, as specified in [RFC3961], the base line is [RFC4121] and the encrypted data per [RFC3961] (which [RFC4121] is based on) is:

```
C1 | H1[1..h]

Where

(C1, newIV) = E(Ke, conf | plaintext | pad, oldstate.ivec)
H1 = HMAC(Ki, conf | plaintext+encrypted-data | pad)
```

where the "plaintext+encrypted-data" is all the input data buffers supplied to GSS_WrapEx() concatenated in the order provided in the ordered list, input_message.

The RRC field described in section 4.2.5 of [RFC4121] is 12 if no encryption is requested or 16 if encryption is requested. The RRC field is chosen such that all the data can be encrypted in place. The trailing meta-data H1 is rotated 12 or 16 octets (based on the RRC field) as specified in section 4.2.5 of [RFC4121]. Thus the token buffer contains the header described in section 4.2.6.2 of [RFC4121] with the rotated H1 that is placed before the encrypted confounder and after the header.

If the session key encryption type is DES-CBC-MD5 or DES-CBC-CRC per [RFC3961]:

- The base line is [RFC1964].
- The ordered list contains the header described in section 1.2.2 of [RFC1964].
- The data is encrypted in place.

The "to-be-signed data" in section 1.2.2.1 of [RFC1964] is a concatenation of all the input_message data for which sign==TRUE. Only the input data with encrypt set to TRUE is encrypted in output_message. The InitialContextToken header as specified in section 1.1 of [RFC1964]] is included at the beginning of the ordered list.

For [MS-RPCE], the length field in the above pseudo ASN.1 header does not include the length of the concatenated data if [RFC1964] is used.

If the session key encryption type is RC4-HMAC or RC4-HMAC-EXP per [RFC3961]:

- The base line is [RFC4757].
- The ordered list contains the header described in section 7.3 of [RFC4757].
- The data (excluding the conf_req_flag set to FALSE) is encrypted in place.

The "to-be-signed data" in section 7.3 of [RFC4757] is a concatenation of all the input buffers for which sign==TRUE. The InitialContextToken pseudo ASN.1 header is included at the beginning of the token header.

3.4.5.5 GSS_UnwrapEx() Call

This call is an extension to GSS_Unwrap ([RFC2743] section 2.3.4) that passes multiple buffers.

Inputs:

- context_handle CONTEXT HANDLE

- input_message ORDERED LIST of:
  - conf_state BOOLEAN
  - signed BOOLEAN
  - data OCTET STRING
- signature OCTET STRING

Outputs:

- qop_req INTEGER, -- 0 specifies default QOP
- major_status INTEGER
- minor_status INTEGER
- output_message ORDERED LIST (in same order as input_message) of:
  - conf_state BOOLEAN
  - data OCTET STRING

This call is identical to GSS_Unwrap, except that it supports multiple input buffers. Input data buffers for which conf_state==TRUE are decrypted in output_message. The signature is verified for the input data buffers where signed==TRUE, that are concatenated as specified in section 3.4.5.4.1.

3.4.5.6 GSS_GetMICEx() Call

Inputs:

- context_handle CONTEXT HANDLE
- qop_req INTEGER, -- 0 specifies default QOP
- message ORDERED LIST of:
  - sign BOOLEAN
  - data OCTET STRING

Outputs:

- major_status INTEGER
- minor_status INTEGER
- message ORDERED LIST of:
  - signed BOOLEAN
  - data OCTET STRING
- per_msg_token OCTET STRING

This call is identical to GSS_GetMIC, except that it supports multiple input buffers. Input data buffers where sign==TRUE are concatenated together and the resulting OCTET STRING is signed as specified by the following RFCs, depending on the session key encryption type:

- DES-CBC-MD5 or DES-CBC-CRC [RFC1964] [RFC3961]
- RC4-HMAC or RC4-HMAC-EXP per [RFC3961] [RFC4757]
- AES128-CTS-HMAC-SHA1-96 or AES256-CTS-HMAC-SHA1-96 [RFC3961] [RFC4121]

3.4.5.7 GSS_VerifyMICEx() Call

Inputs:

- context_handle CONTEXT HANDLE
- message ORDERED LIST of:
  - signed BOOLEAN
  - data OCTET STRING
- per_msg_token OCTET STRING

Outputs:

- qop_state INTEGER
- major_status INTEGER
- minor_status INTEGER

This call is identical to GSS_VerifyMIC, except that it supports multiple input buffers. Input data buffers where signed==TRUE are concatenated together and the signature is verified against the resulting concatenated buffer.

3.4.6 Timer Events

No timer events exist other than those specified in [RFC4120].

3.4.7 Other Local Events

There are no other local events except what is driven by the application layer protocol.

4 Protocol Examples

The following sections describe two common scenarios to illustrate the function of the KILE.

4.1 Interactive Logon Using Passwords

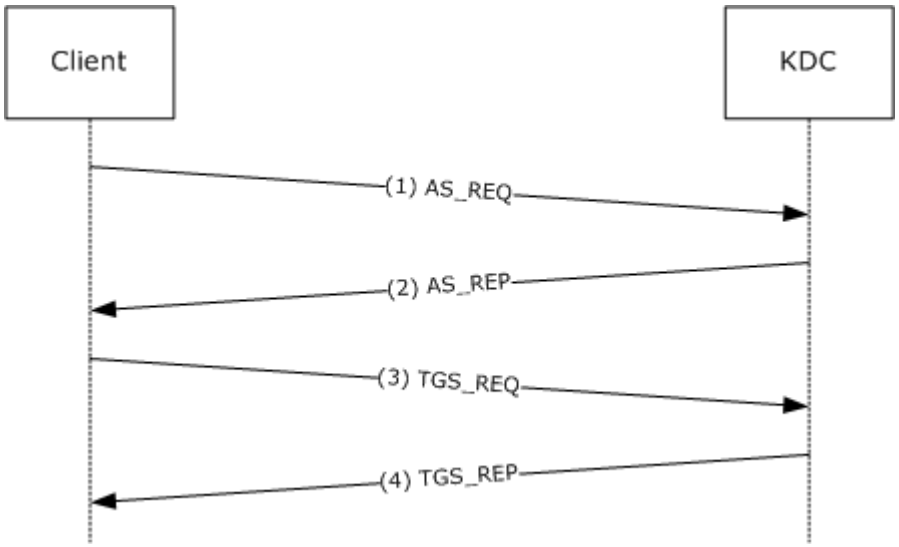

Figure 2: Interactive logon that uses passwords

Step 1: A user who attempts to log on to a client, types a password at the logon screen, and an AS-REQ for a ticket-granting ticket (TGT) is generated. The AS-REQ, which uses the user name and password, is sent to the Key Distribution Center (KDC).

Step 2: In response to receiving the AS-REQ for a TGT, the KDC authenticates the user by checking that the credentials that are used in the AS-REQ are the same as that of the user's ([RFC4120] section 3.1). The KDC builds an AS-REP from the TGT and other requisite data, and sends it back to the client.

The KDC builds a PAC (section 3.2.5.2). Data in the PAC includes account data for the user that is used for logging onto the client. The account data is expected to be supplied by the KDC that queries an account service for the account data. The KDC inserts the PAC that contains the account data that is received from the account service into the authorization_data field of the TGT.

Step 3: The client then sends a TGS-REQ based on the TGT that is obtained in step 2 to obtain a service ticket that is formatted according to the Kerberos protocol for completing a logon process at the local workstation. The client runtime issues a request to host/hostname.domain, where hostname is the actual name of the client machine, and domain is the domain or realm of the client machine.

Step 4: The KDC responds to the TGS-REQ with a TGS-REP that contains the service ticket for the local workstation. The authorization data from step 2 is carried forward to the service ticket, with additional group processing (section 3.2.5.3). The service ticket is then interpreted by the Kerberos runtime within the local workstation.

The following fields from the PAC ([MS-PAC] is the authoritative reference for formatting and encoding these fields) are required by the Kerberos interactive logon to authorize the user for local logon, and to establish the necessary management profile for the user:

- **LogonTime**: The time when the user last logged on. This field is an absolute-format Windows standard time value.
- **LogoffTime**: The time when the user should log off. This field is an absolute-format Windows standard time value.
- **KickOffTime**: The time when the system forces the user to log off. This field is an absolute-format Windows standard time value. Note that Windows users are not forced to log off interactively; however, their network connections may be closed.
- **PasswordLastSet**: The time and date that the password was last changed. This field is an absolute format Windows standard time value.
- **PasswordCanChange**: The time and date when the user is reminded to change passwords. This field is an absolute-format Windows standard time value.
- **EffectiveName**: The text field that contains the effective name of the account that is validated by Active Directory.
- **FullName**: The text field that contains the user's full name.
- **LogonScript**: The text field that contains the relative path to the account's logon script.
- **ProfilePath**: The text field that contains the path to a user's roaming profile. This field is only used if the user has a roaming profile.
- **HomeDirectory**: The text field that contains the user's home directory.
- **HomeDirectoryDrive**: The text field that contains the drive that contains the user's home directory.
- **LogonCount**: The number of times the user is currently logged on.
- **BadPasswordCount**: The number of times a bad password was applied to the account since the last successful logon.
- **LogonServer**: The text field that contains the name of the server that processed the logon request.
- **LogonDomainName**: The text field that contains the name of the computer that is making the account logon request.
- **UserAccountControl**: Flags that control the behavior of the user account.

4.2 Network Logon

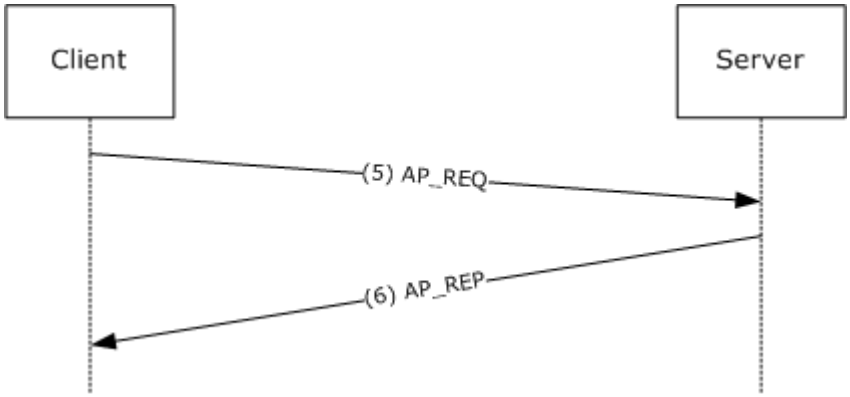

Figure 3: Network Logon

When an application wants to use Kerberos-based authentication, it uses either the higher-level SSPI API to invoke Kerberos directly; or it uses SPNEGO [MS-SPNG], which in turn invoke Kerberos.

This may cause steps 1 to 4 (section 4.1) to be repeated if there are new credentials supplied. It may also cause steps 3 and 4 (section 4.1) to be repeated if the server has not previously cached a ticket for the client.

Step 5: When the service ticket to the application server is obtained, the client authenticates itself to the server by sending an AP-REQ wrapped in Generic Security Services (GSS) formatting (section 3.3 and [RFC1964]).

Step 6: The Kerberos runtime on the server validates the ticket by decrypting it and it validates the authenticator by decrypting and checking for replay and other attacks ([RFC4120] section 3.2).

Invoking the Kerberos runtime to authenticate a **session** is typically done through the SSPI API. Higher-level constructs, for example, remote file access, can also trigger the connection. After the server-side Kerberos runtime validates the ticket and authenticator, it makes the authorization data from the ticket available to the service, typically through a Windows-specific object that is known as an access token, which is used with the Windows system-provided authorization functions.

5 Security

Older versions of MIT Kerberos do not support RC4 and therefore, the only common option for interoperability is DES. To obtain the security benefits of a stronger 128-bit key, upgrade to the latest version of MIT Kerberos.

Other general Kerberos security considerations are specified in [RFC4120] section 10.

5.1 Security Considerations for Implementers

There are no security considerations for implementers.

5.2 Index of Security Parameters

There are no security parameters for this protocol extension.

6 Appendix A: Windows Behavior

The information in this specification is applicable to the following versions of Windows:

- Windows NT
- Windows 2000
- Windows XP
- Windows Server 2003
- Windows Vista
- Windows Server 2008

Exceptions, if any, are noted below. Unless otherwise specified, any statement of optional behavior in this specification prescribed using the terms SHOULD or SHOULD NOT implies Windows behavior in accordance with the SHOULD or SHOULD NOT prescription. Unless otherwise specified, the term MAY implies that Windows does not follow the prescription.

<1> Section 1.4: The equivalent to the GSS-API layer in Windows is called the Security Support Provider Interface (SSPI). For more information, see [SSPI]. In brief, Windows applications use the SSPI interface to authenticate communications. The SSPI layer adds the appropriate protocol-specific authentication messages.

<2> Section 1.5: Configuration options can modify this setting. Windows uses the Network Time Protocol and Authentication Extensions [MS-SNTP], for synchronization of the time between the three parties

<3> Section 1.5: The Windows implementation of the KDC is integrated with Active Directory (AD). All accounts and related credentials are stored in AD; KILE does not provide a distinct account database.

<4> Section 2.1: The default values for the message size threshold are shown below for different versions of Windows.

Windows version	Message size
Windows 2000 (initial release) –Windows 2000 SP3	2000 bytes
Windows 2000 SP4	1465 bytes
Windows XP (initial release) and Windows XP SP1	2000 bytes
Windows XP SP2	1500 bytes
Windows Server 2003 (initial release), Windows XP 64-Bit Edition, Windows Server 2003 SP1, Windows Server 2003 R2, and Windows Vista	1465 bytes

Note: Windows NT does not include a Kerberos implementation.

<5> Section 2.1: Windows can support manual configuration of KDCs; however, this is not recommended. KDCs publish SRV records that describe their capabilities to a DNS server that supports dynamic DNS. They also produce a standard DNS file that can be incorporated into static

DNS servers. Domain controllers publish more than just the KDC SRV records. For more information, see [MS-ADTS] section 7.3.

<6> Section 2.2.2: The **LSAP_TOKEN_INFO_INTEGRITY** structure is supported in Windows Vista SP1 and Windows Server 2008.

<7> Section 2.2.3: The KERB-AD-RESTRICTION-ENTRY structure is supported in Windows Vista SP1 and Windows Server 2008.

<8> Section 3.1.1.3: Windows has a ticket cache and makes the ticket cache available to client applications at their request. Programmatic methods for querying the contents, purging the contents, or purging individual tickets are also available.

In Windows Server 2003, Windows Vista and Windows Server 2008, TGTs are automatically renewed. Renewal attempts begin at 10 minutes prior to expiration, unless the renewUntil time of the TGT is less than five minutes in the future.

<9> Section 3.1.5.1: Windows clients by default will request a PAC.

<10> Section 3.1.5.2: Windows 2000KDCs select the encryption type based on the preference order in the client request. In Windows Server 2003, Windows Vista and Windows Server 2008, KDCs select the strongest encryption type first, regardless of the preference expressed by the client

<11> Section 3.1.5.2: Supported in Windows Vista and Windows Server 2008.

<12> Section 3.1.5.2: Supported in Windows Vista and Windows Server 2008.

<13> Section 3.1.5.5: IPv6 addresses are supported in Windows Vista and Windows Server 2008 but not in prior versions of Windows.

<14> Section 3.1.5.6: Active Directory implements a technique known as "diacritical folding" (for more information, see [ADFOLDING]). Active Directory, and the Windows server operating systems in general, are intended for deployment in worldwide networks. As such, not all nodes will have input devices that allow for entering appropriate diacritical or similar marks. The goal is for "Götz" to log on, even if the keyboard only allows "Gotz." Diacritical folding allows matching of names that differ only in a set of diacritical marks. This is accomplished by Active Directory using the following flags with the **CompareString** function: NORM_IGNORECASE, NORM_IGNOREKANATYPE, NORM_IGNORENONSPACE, NORM_IGNOREWIDTH. Note that this applies only to names; passwords (and the transformation of a password to a key) are governed by the actual key generation specification ([RFC4120], [RFC4757], and [RFC3962]).

<15> Section 3.1.5.7: Supported in Windows Server 2008.

<16> Section 3.1.5.9: Windows clients explicitly request a PAC be included.

<17> Section 3.2.1.1: Windows exposes these logical parameters through the Security SSPI on Windows.

<18> Section 3.2.3: The locator, also called a DC locator, is responsible for dynamic location by using SRV records. For more information, see [MS-ADTS] section 7.3. Using the DNS-based discovery method ([MS-ADTS] section 7.3.6), the first DC that produces a valid response is discovered as the DC which is the KDC used for Kerberos authentication. If no KDC can be found, KILE cannot continue.

<19> Section 3.2.3: Windows can be configured with a list of KDCs for a realm for interoperability purposes through the use of the Windows registry and Windows-specific configuration tools.

<20> Section 3.2.4.1: From the user's PAC, Windows identifies the user's principal ID and group memberships and populates the user's security token with that information. The information is subsequently used in access control decisions.

<21> Section 3.2.4.1: All Windows environments support this behavior. The user principal name (UPN) is nominally treated as a general string by the KDC, which devolves the resolution of the name to Active Directory[MS-ADTS]. The administrator of Active Directory can determine the allowable suffixes to the UPN. For Windows to support the usage of the apparent domain "example.com", Active Directory MUST have been configured to allow "example.com" as a valid UPN suffix.

<22> Section 3.2.5.3: Supported in Windows Vista SP1 and Windows Server 2008.

<23> Section 3.2.6: Windows client implementations include configured values for the initial time-out, T, and an increase factor on that time-out, I. The first time-out is T, the second is T+I, the third is T+2I. Windows clients try three times to connect to a KDC and after three failures declare the AS-REQ or TGS-REQ message a failure.

Subsequent behavior depends on how many domain controllers are known or how many have been attempted to be discovered, among other factors, but is not timer-related.

Since AP exchanges are carried as a field in an application protocol message, that application defines the time-out behavior for its message and therefore the AP exchanges. This is not defined as part of either [RFC4120] or KILE.

<24> Section 3.3.1: Windows has a setting for the lifetime for service tickets and a separate setting for the lifetime of TGTs. Both are 10 hours by default.

<25> Section 3.3.1: Windows uses Directory Service policy settings.

<26> Section 3.3.1.1: The AD maintains this flag [0x80000] in the userAccountControl attribute ([MS-ADA3] section 2.341).

<27> Section 3.3.1.1: The AD maintains this flag [0x4000] in the userAccountControl attribute ([MS-ADA3] section 2.341).

<28> Section 3.3.1.1: The AD maintains this flag [0x10000] in the userAccountControl attribute ([MS-ADA3] section 2.341).

<29> Section 3.3.1.1: The AD maintains this flag [0x40000] in the userAccountControl attribute ([MS-ADA3] section 2.341).

<30> Section 3.3.1.1: The AD maintains this flag [0x2000] in the userAccountControl attribute ([MS-ADA3] section 2.341).

<31> Section 3.3.1.1: The AD maintains this flag [0x8000] in the userAccountControl attribute ([MS-ADA3] section 2.341).

<32> Section 3.3.3: Windows uses AD as the store for account information. The AD is responsible for replicating information from one node to another.

When a DC starts up, the KDC is not brought online until AD has replicated with its peers. The KDC queries LsarQueryInformationPolicy ([MS-LSAD] section 3.1.4.4.3) with the following value for the InformationClass parameter:

- PolicyAccountDomainInformation: for the domain name and domain SID.

The KDC queries LsarQueryDomainInformationPolicy ([MS-LSAD] section 3.1.4.4.7) with the following value for the InformationClass parameter:

- PolicyDomainKerberosTicketInformation for MaxServiceTicketAge, MaxTicketAge, MaxRenewAge, MaxClockSkew, MaxServiceTicketAge.

This startup sequence ensures that a KDC does not attempt to issue tickets with old keys.

<33> Section 3.3.5.1: Windows 2000 KDCs will canonicalize the name in the resulting ticket, based on the name of the account that is ultimately used in AD.

Windows Server 2003 KDCs do not honor the canonicalize flag except for referrals [Referrals], and they do not perform any canonicalization.

<34> Section 3.3.5.2: Supported in Windows Vista SP1 and Windows Server 2008.

<35> Section 3.3.5.4.4: Supported in Windows Server 2003 and Windows Server 2008.

<36> Section 3.3.5.4.4: The AD supports an attribute for accounts to list the services for which the account can use forwarded tickets.

<37> Section 3.4.5.3: Supported in Windows Vista SP1 and Windows Server 2008.

7 Index