Generating KDC Referrals to Locate Kerberos Realms

draft-ietf-krb-wg-kerberos-referrals-10

Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with Section 6 of BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

This Internet-Draft will expire on August 28, 2008.

Copyright Notice

Copyright (C) The IETF Trust (2008).

Abstract

The memo documents a method for a Kerberos Key Distribution Center (KDC) to respond to client requests for Kerberos tickets when the client does not have detailed configuration information on the realms of users or services. The KDC will handle requests for principals in other realms by returning either a referral error or a cross-realm TGT to another realm on the referral path. The clients will use this referral information to reach the realm of the target principal and
then receive the ticket.

Table of Contents

1. Introduction ........................................... 3
2. Conventions Used in This Document ..................... 4
3. Requesting a Referral .................................. 4
4. Realm Organization Model ............................... 5
5. Enterprise Principal Name Type ........................ 5
6. Name Canonicalization ................................... 5
7. Client Referrals ......................................... 7
8. Server Referrals ......................................... 9
9. Cross Realm Routing .................................... 11
10. Caching Information .................................... 11
11. Open Issues ............................................ 12
12. Number Assignments .................................... 12
13. Security Considerations ............................... 13
   13.1. Shared-password case .............................. 13
14. Acknowledgments ....................................... 14
15. References .............................................. 14
   15.1. Normative References .............................. 14
   15.2. Informative References ............................ 14
Appendix A. Compatibility with Earlier Implementations of Name Canonicalization ......................... 14
Appendix B. Document history [REMOVE BEFORE PUBLICATION] .. 16
Authors’ Addresses .......................................... 16
Intellectual Property and Copyright Statements ............ 18
1. Introduction

Current implementations of the Kerberos AS and TGS protocols, as defined in [RFC4120], use principal names constructed from a known user or service name and realm. A service name is typically constructed from a name of the service and the DNS host name of the computer that is providing the service. Many existing deployments of Kerberos use a single Kerberos realm where all users and services would be using the same realm. However in an environment where there are multiple trusted Kerberos realms, the client needs to be able to determine what realm a particular user or service is in before making an AS or TGS request. Traditionally this requires client configuration to make this possible.

When having to deal with multiple trusted realms, users are forced to know what realm they are in before they can obtain a ticket granting ticket (TGT) with an AS request. However, in many cases the user would like to use a more familiar name that is not directly related to the realm of their Kerberos principal name. A good example of this is an RFC 822 style email name. This document describes a mechanism that would allow a user to specify a user principal name that is an alias for the user’s Kerberos principal name. In practice this would be the name that the user specifies to obtain a TGT from a Kerberos KDC. The user principal name no longer has a direct relationship with the Kerberos principal or realm. Thus the administrator is able to move the user’s principal to other realms without the user having to know that it happened.

Once a user has a TGT, they would like to be able to access services in any trusted Kerberos realm. To do this requires that the client be able to determine what realm the target service principal is in before making the TGS request. Current implementations of Kerberos typically have a table that maps DNS host names to corresponding Kerberos realms. The user-supplied host name or its domain component is looked up in this table (often using the result of some form of host name lookup performed with insecure DNS queries, in violation of [RFC4120]). The corresponding realm is then used to complete the target service principal name.

This traditional mechanism requires that each client have very detailed configuration information about the hosts that are providing services and their corresponding realms. Having client side configuration information can be very costly from an administration point of view - especially if there are many realms and computers in the environment.

There are also cases where specific DNS aliases (local names) have been setup in an organization to refer to a server in another
organization (remote server). The server has different DNS names in each organization and each organization has a Kerberos realm that is configured to service DNS names within that organization. Ideally users are able to authenticate to the server in the other organization using the local server name. This would mean that the local realm be able to produce a ticket to the remote server under its name. The administrator in the local realm could give that remote server an identity in the local realm and then have that remote server maintain a separate secret for each alias it is known as. Alternatively the administrator could arrange to have the local realm issue a referral to the remote realm and notify the requesting client of the server’s remote name that should be used in order to request a ticket.

This memo proposes a solution for these problems and simplifies administration by minimizing the configuration information needed on each computer using Kerberos. Specifically it describes a mechanism to allow the KDC to handle canonicalization of names, provide for principal aliases for users and services and allow the KDC to determine the trusted realm authentication path by being able to generate referrals to other realms in order to locate principals.

Two kinds of KDC referrals are introduced in this memo:

1. Client referrals, in which the client doesn’t know which realm contains a user account.
2. Server referrals, in which the client doesn’t know which realm contains a server account.

### 2. Conventions Used in This Document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

### 3. Requesting a Referral

In order to request referrals defined in section 5, 6, and 7, the Kerberos client MUST explicitly request the canonicalize KDC option (bit 15) [RFC4120] for the AS-REQ or TGS-REQ. This flag indicates to the KDC that the client is prepared to receive a reply that contains a principal name other than the one requested.

```
KDCOptions ::= KerberosFlags
               -- canonicalize (15)
```

The client should expect, when sending names with the "canonicalize"
KDC option, that names in the KDC’s reply MAY be different than the
name in the request. A referral TGT is a cross realm TGT that is
returned with the server name of the ticket being different from the
server name in the request [RFC4120].

4. Realm Organization Model

This memo assumes that the world of principals is arranged on
multiple levels: the realm, the enterprise, and the world. A KDC may
issue tickets for any principal in its realm or cross-realm tickets
for realms with which it has a direct trust relationship. The KDC
also has access to a trusted name service that can resolve any name
from within its enterprise into a realm. This trusted name service
removes the need to use an un-trusted DNS lookup for name resolution.

For example, consider the following configuration, where lines
indicate trust relationships:

```
EXAMPLE.COM
  /   \
  /     \
ADMIN.EXAMPLE.COM  DEV.EXAMPLE.COM
```

In this configuration, all users in the EXAMPLE.COM enterprise could
have principal names such as alice@EXAMPLE.COM, with the same realm
portion. In addition, servers at EXAMPLE.COM should be able to have
DNS host names from any DNS domain independent of what Kerberos realm
their principals reside in.

5. Enterprise Principal Name Type

The NT-ENTERPRISE type principal name contains one component, a
string of realm-defined content, which is intended to be used as an
alias for another principal name in some realm in the enterprise. It
is used for conveying the alias name, not for the real principal
names within the realms, and thus is only useful when name
canonicalization is requested.

6. Name Canonicalization

A service or account may have multiple principal names. More useful,
though, is a globally unique name that allows unification of email
and security principal names. For example, all users at EXAMPLE.COM may have a client principal name of the form "joe@EXAMPLE.COM" even though the principals are contained in multiple realms. This global name is again an alias for the true client principal name, which indicates what realm contains the principal. Thus, accounts "alice" in the realm DEV.EXAMPLE.COM and "bob" in ADMIN.EXAMPLE.COM may log on as "alice@EXAMPLE.COM" and "bob@EXAMPLE.COM".

This utilizes a new client principal name type, as the AS-REQ message only contains a single realm field, and the realm portion of this name corresponds to the Kerberos realm with which the request is made. Thus, the entire name "alice@EXAMPLE.COM" is transmitted as a single component in the client name field of the AS-REQ message, with a name type of NT-ENTERPRISE [RFC4120] (and the local realm name). The KDC will recognize this name type and then transform the requested name into the true principal name if the client account resides in the local realm. The true principal name can have a name type different from the requested name type. Typically the true principal name will be a NT-PRINCIPAL [RFC4120].

If the "canonicalize" KDC option is set, then the KDC MAY change the client principal name and type in the AS response and ticket returned from the name type of the client name in the request, and include a mandatory PA-DATA object authenticating the client name mapping:

```
ReferralInfo ::= SEQUENCE {
  requested-name  [0] PrincipalName,
  mapped-name     [1] PrincipalName,
  ...
}
PA-CLIENT-CANONICALIZED ::= SEQUENCE {
  names          [0] ReferralInfo,
  canon-checksum [1] Checksum
}
```

The canon-checksum field is computed over the DER encoding of the names sequences, using the AS reply key and a key usage value of (TBD).

If the client name is unchanged, the PA-CLIENT-CANONICALIZED data is not included. If the client name is changed, and the PA-CLIENT-CANONICALIZED field does not exist, or the checksum cannot be verified, or the requested-name field doesn’t match the client name in the originally-transmitted request, the client should discard the response.

For example the AS request may specify a client name of "bob@EXAMPLE.COM" as an NT-ENTERPRISE name with the "canonicalize" KDC
option set and the KDC will return with a client name of "104567" as a NT-UID, and a PA-CLIENT-CANONICALIZED field listing the NT-ENTERPRISE "bob@EXAMPLE.COM" principal as the requested-name and the NT-UID "104567" principal as the mapped-name.

(It is assumed that the client discovers whether the KDC supports the NT-ENTERPRISE name type via out of band mechanisms.)

In order to enable one party in a user-to-user exchange to confirm the identity of another when only the alias is known, the KDC MAY include the following authorization data element, wrapped in AD-KDC-ISSUED, in the initial credentials and copy it from a ticket-granting ticket into additional credentials:

AD-LOGIN-ALIAS ::= SEQUENCE { -- ad-type number TBD --
   login-aliases [0] SEQUENCE(1..MAX) OF PrincipalName,
}

The login-aliases field lists one or more of the aliases the principal may have used in the initial ticket request.

The recipient of this authenticator must check the AD-LOGIN-ALIAS names, if present, in addition to the normal client name field, against the identity of the party with which it wishes to authenticate; either should be allowed to match. (Note that this is not backwards compatible with [RFC4120]; if the server side of the user-to-user exchange does not support this extension, and does not know the true principal name, authentication may fail if the alias is sought in the client name field.)

The use of AD-KDC-ISSUED authorization data elements in cross-realm cases has not been well explored at this writing; hence we will only specify the inclusion of this data in the one-realm case. The alias information should be dropped in the general cross-realm case. However, a realm may implement a policy of accepting and re-signing (wrapping in a new AD-KDC-ISSUED element) alias information provided by certain other realms in the cross-realm ticket-granting service.

7. Client Referrals

The simplest form of ticket referral is for a user requesting a ticket using an AS-REQ. In this case, the client machine will send the AS-REQ to a convenient trusted realm, for example the realm of the client machine. In the case of the name alice@EXAMPLE.COM, the client MAY optimistically choose to send the request to EXAMPLE.COM. The realm in the AS-REQ is always the name of the realm that the request is for as specified in [RFC4120].
The KDC will try to lookup the name in its local account database. If the account is present in the realm of the request, it SHOULD return a KDC reply structure with the appropriate ticket.

If the account is not present in the realm specified in the request and the "canonicalize" KDC option is set, the KDC will try to lookup the entire name, alice@EXAMPLE.COM, using a name service. If this lookup is unsuccessful, it MUST return the error KDC_ERR_C_PRINCIPAL_UNKNOWN [RFC4120]. If the lookup is successful, it MUST return an error KDC_ERR_WRONG_REALM [RFC4120] and in the error message the crealm field will contain either the true realm of the client or another realm that MAY have better information about the client’s true realm. The client SHALL NOT use a cname returned from a Kerberos error until that name is validated.

If the client receives a KDC_ERR_WRONG_REALM error, it will issue a new AS request with the same client principal name used to generate the first referral to the realm specified by the realm field of the Kerberos error message corresponding to the first request. (The client realm name will be updated in the new request to refer to this new realm.) The client SHOULD repeat these steps until it finds the true realm of the client. To avoid infinite referral loops, an implementation should limit the number of referrals. A suggested limit is 5 referrals before giving up.

Since the same client name is sent to the referring and referred-to realms, both realms must recognize the same client names. In particular, the referring realm cannot (usefully) define principal name aliases that the referred-to realm will not know.

The true principal name of the client, returned in AS-REQ, can be validated in a subsequent TGS message exchange where its value is communicated back to the KDC via the authenticator in the PA-TGS-REQ padata [RFC4120]. However, this requires trusting the referred-to realm’s KDCs. Clients should limit the referral mappings they will accept to realms trusted via some local policy. Some possible factors that might be taken into consideration for such a policy might include:

- Any realm indicated by the local KDC, if the returned KRB-ERROR message is protected, for example using a public key known to be associated with the KDC
- A list of realms configured by an administrator
- Any realm accepted by the user when explicitly prompted
8. Server Referrals

The primary difference in server referrals is that the KDC MUST return a referral TGT rather than an error message as is done in the client referrals. There needs to be a place to include in the reply information about what realm contains the server. This is done by returning information about the server name in the pre-authentication data field of the KDC reply [RFC4120], as specified later in this section.

If the KDC resolves the server principal name into a principal in the realm specified by the service realm name, it will return a normal ticket.

If the "canonicalize" flag in the KDC options is not set, the KDC MUST only look up the name as a normal principal name in the specified server realm. If the "canonicalize" flag in the KDC options is set and the KDC doesn’t find the principal locally, the KDC MAY return a cross-realm ticket granting ticket to the next hop on the trust path towards a realm that may be able to resolve the principal name. The true principal name of the server SHALL be returned in the padata of the reply if it is different from what is specified the request.

When a referral TGT is returned, the KDC MUST return the target realm for the referral TGT as an KDC supplied pre-authentication data element in the response. This referral information in pre-authentication data MUST be encrypted using the session key from the reply ticket. The key usage value for the encryption operation used by PA-SERVER-REFERRAL is 26.

The pre-authentication data returned by the KDC, which contains the referred realm and the true principal name of server, is encoded in DER as follows.
PA-SERVER-REFERRAL 25

PA-SERVER-REFERRAL-DATA ::= EncryptedData
   -- ServerReferralData --

ServerReferralData ::= SEQUENCE {
   referred-realm           [0] Realm OPTIONAL,
      -- target realm of the referral TGT
   true-principal-name      [1] PrincipalName OPTIONAL,
      -- true server principal name
   requested-principal-name [2] PrincipalName OPTIONAL,
      -- requested server name
   referral-valid-until     [3] KerberosTime OPTIONAL,
      ...}

Clients SHALL NOT accept a reply ticket in which the server principal name is different from that of the request, if the KDC response does not contain a PA-SERVER-REFERRAL padata entry.

The requested-principal-name MUST be included by the KDC, and MUST be verified by the client, if the client sent an AS-REQ, as protection against a man-in-the-middle modification to the AS-REQ message.

The referred-realm field is present if and only if the returned ticket is a referral TGT, not a service ticket for the requested server principal.

When a referral TGT is returned and the true-principal-name field is present, the client MUST use that name in the subsequent requests to the server realm when following the referral.

Client SHALL NOT accept a true server principal name for a service ticket if the true-principal-name field is not present in the PA-SERVER-REFERRAL data.

The client will use this referral information to request a chain of cross-realm ticket granting tickets until it reaches the realm of the server, and can then expect to receive a valid service ticket.

However an implementation should limit the number of referrals that it processes to avoid infinite referral loops. A suggested limit is 5 referrals before giving up.

The client may cache the mapping of the requested name to the name of the next realm to use and the principal name to ask for. (See Section 10.) The referral-valid-until field, if present, conveys how long this information is valid for.
Here is an example of a client requesting a service ticket for a service in realm DEV.EXAMPLE.COM where the client is in ADMIN.EXAMPLE.COM.

+NC = Canonicalize KDCOption set
+PA-REFERRAL = returned PA-SERVER-REFERRAL
C: TGS-REQ sname=http/foo.dev.example.com +NC to ADMIN.EXAMPLE.COM
S: TGS-REP sname=krbtgt/EXAMPLE.COM@ADMIN.EXAMPLE.COM +PA-REFERRAL
    containing EXAMPLE.COM as the referred realm with no true-principal-name
C: TGS-REQ sname=http/foo.dev.example.com +NC to EXAMPLE.COM
S: TGS-REP sname=krbtgt/DEV.EXAMPLE.COM@EXAMPLE.COM +PA-REFERRAL
    containing DEV.EXAMPLE.COM as the referred realm with no true-principal-name
C: TGS-REQ sname=http/foo.dev.example.com +NC to DEV.EXAMPLE.COM
S: TGS-REP sname=http/foo.dev.example.com@DEV.EXAMPLE.COM

Note that any referral or alias processing of the server name in user-to-user authentication should use the same data as client name canonicalization or referral. Otherwise, the name used by one user to log in may not be useable by another for user-to-user authentication to the first.

9. Cross Realm Routing

The current Kerberos protocol requires the client to explicitly request a cross-realm TGT for each pair of realms on a referral chain. As a result, the client need to be aware of the trust hierarchy and of any short-cut trusts (those that aren’t parent-child trusts).

Instead, using the server referral routing mechanism as defined in Section 8, The KDC will determine the best path for the client and return a cross-realm TGT as the referral TGT, and the target realm for this TGT in the PA-SERVER-REFERRAL of the KDC reply.

If the "canonicalize" KDC option is not set, the KDC SHALL NOT return a referral TGT. Clients SHALL NOT process referral TGTs if the KDC response does not contain the PA-SERVER-REFERRAL padata.

10. Caching Information

It is possible that the client may wish to get additional credentials for the same service principal, perhaps with different authorization-data restrictions or other changed attributes. The return of a server referral from a KDC can be taken as an indication that the
requested principal does not currently exist in the local realm. Clearly, it would reduce network traffic if the clients could cache that information and use it when acquiring the second set of credentials for a service, rather than always having to re-check with the local KDC to see if the name has been created locally.

If the referral-valid-until field is provided in the PA-SERVER-REFERRAL-DATA message, it indicates the expiration time of this data; if it is not included, the expiration time of the TGT is used. When the TGT expires, the previously returned referral from the local KDC should be considered invalid, and the local KDC must be asked again for information for the desired service principal name. (Note that the client may get back multiple referral TGTs from the local KDC to the same remote realm, with different lifetimes. The lifetime information must be properly associated with the requested service principal names. Simply having another TGT for the same remote realm does not extend the validity of previously acquired information about one service principal name.) If the client is still in contact with the service and needs to reauthenticate to the same service regardless of local service principal name assignments, it should use the referred-realm and true-principal-name values when requesting new credentials.

Accordingly, KDC authors and maintainers should consider what factors (e.g., DNS alias lifetimes) they may or may not wish to incorporate into credential expiration times in cases of referrals.

11. Open Issues

Client referral info validation

When should client name aliases be included in credentials? Should all known client name aliases be included, or only the one used at initial ticket acquisition?

Should list the policies that need to be defined.

More examples: u2u, policy checks, maybe cross-realm.

Restore server name canonicalization from early drafts.

12. Number Assignments

Most number registries in the Kerberos protocol have not been turned over to IANA for management at the time of this writing, hence this is not an "IANA Considerations" section.
Various values do need assigning for this draft:
AD-LOGIN-ALIAS
PA-CLIENT-CANONICALIZED
key usage value for PA-CLIENT-CANONICALIZED field canon-checksum

13. Security Considerations

For the AS exchange case, it is important that the logon mechanism not trust a name that has not been used to authenticate the user. For example, the name that the user enters as part of a logon exchange may not be the name that the user authenticates as, given that the KDC_ERR_WRONG_REALM error may have been returned. The relevant Kerberos naming information for logon (if any), is the client name and client realm in the service ticket targeted at the workstation that was obtained using the user’s initial TGT.

How the client name and client realm is mapped into a local account for logon is a local matter, but the client logon mechanism MUST use additional information such as the client realm and/or authorization attributes from the service ticket presented to the workstation by the user, when mapping the logon credentials to a local account on the workstation.

13.1. Shared-password case

A special case to examine is when the user is known (or correctly suspected) to use the same password for multiple accounts. A man-in-the-middle attacker can either alter the request on its way to the KDC, changing the client principal name, or reply to the client with a response previously send by the KDC in response to a request from the attacker. The response received by the client can then be decrypted by the user, though if the default "salt" generated from the principal name is used to produce the user’s key, a PA-ETYPE-INFO or PA-ETYPE-INFO2 preauth record may need to be added or altered by the attacker to cause the client software to generate the key needed for the message it will receive. None of this requires the attacker to know the user’s password, and without further checking, could cause the user to unknowingly use the wrong credentials.

In normal [RFC4120] operation, a generated AP-REQ message includes in the Authenticator field a copy of the client’s idea of its own principal name. If this differs from the name in the KDC-generated Ticket, the application server will reject the message.

With client name canonicalization as described in this document, the client may get its principal name from the response from the KDC. Requiring the PA-CLIENT-CANONICALIZED data lets the client securely
check that the requested name was not altered in transit. If the PA-
CLIENT-CANONICALIZED data is absent, the client can use the principal
name it requested.

14. Acknowledgments

Sam Hartman and authors came up with the idea of using the ticket key
to encrypt the referral data, which prevents cut and paste attack
using the referral data and referral TGTs.

John Brezak, Mike Swift, and Jonathan Trostle wrote the initial
version of this document.

Karthik Jaganathan contributed to earlier versions.

15. References

15.1. Normative References

[ RFC2119 ] Bradner, S., "Key words for use in RFCs to Indicate

Kerberos Network Authentication Service (V5)", RFC 4120,
July 2005.

15.2. Informative References

[ RFC3280 ] Housley, R., Polk, W., Ford, W., and D. Solo, "Internet
X.509 Public Key Infrastructure Certificate and
Certificate Revocation List (CRL) Profile", RFC 3280,
April 2002.

[ RFC4556 ] Zhu, L. and B. Tung, "Public Key Cryptography for Initial
Authentication in Kerberos (PKINIT)", RFC 4556, June 2006.

[XPR] Trostle, J., Kosinovsky, I., and M. Swift, "Implementation
of Crossrealm Referral Handling in the MIT Kerberos
Client", Network and Distributed System Security

Appendix A. Compatibility with Earlier Implementations of Name
Canonicalization

The Microsoft Windows 2000 and Windows 2003 releases included an
earlier form of name-canonicalization [XPR]. Here are the differences:

1) The TGS referral data is returned inside of the KDC message as "encrypted pre-authentication data".

\[
\text{EncKDCRepPart ::= SEQUENCE {}
\text{  \quad key [0] EncryptionKey,}
\text{  \quad last-req [1] LastReq,}
\text{  \quad nonce [2] UInt32,}
\text{  \quad key-expiration [3] KerberosTime OPTIONAL,}
\text{  \quad flags [4] TicketFlags,}
\text{  \quad authtime [5] KerberosTime,}
\text{  \quad starttime [6] KerberosTime OPTIONAL,}
\text{  \quad endtime [7] KerberosTime,}
\text{  \quad renew-till [8] KerberosTime OPTIONAL,}
\text{  \quad srealm [9] Realm,}
\text{  \quad sname [10] PrincipalName,}
\text{  \quad caddr [11] HostAddresses OPTIONAL,}
\text{  \quad encrypted-pa-data [12] SEQUENCE OF PA-DATA OPTIONAL}
\} }
\]

2) The preauth data type definition in the encrypted preauth data is as follows:

\[
\text{PA-SVR-REFERRAL-INFO 20}
\]

\[
\text{PA-SVR-REFERRAL-DATA ::= SEQUENCE {}
\text{  \quad referred-name [1] PrincipalName OPTIONAL,}
\text{  \quad referred-realm [0] Realm}
\}}
\]

3) When PKINIT ([RFC4556]) is used, the NT-ENTERPRISE client name is encoded as a Subject Alternative Name (SAN) extension [RFC3280] in the client’s X.509 certificate. The type of the otherName field for this SAN extension is AnotherName [RFC3280]. The type-id field of the type AnotherName is id-ms-sc-logon-upn (1.3.6.1.4.1.311.20.2.3) and the value field of the type AnotherName is a KerberosString [RFC4120]. The value of this KerberosString type is the single component in the name-string [RFC4120] sequence for the corresponding NT-ENTERPRISE name type.

In Microsoft’s current implementation through the use of global catalogs any domain in one forest is reachable from any other domain.
in the same forest or another trusted forest with 3 or less referrals. A forest is a collection of realms with hierarchical trust relationships: there can be multiple trust trees in a forest; each child and parent realm pair and each root realm pair have bidirectional transitive direct trust relationships between them.

While we might want to permit multiple aliases to exist and even be reported in AD-LOGIN-ALIAS, the Microsoft implementation permits only one NT-ENTERPRISE alias to exist, so this question had not previously arisen.

Appendix B. Document history [REMOVE BEFORE PUBLICATION]

10 TBD
09 Changed to EXAMPLE.COM instead of using Morgan Stanley’s domain. Rewrote description of existing practice. (Don’t name the lookup table consulted. Mention that DNS "canonicalization" is contrary to [RFC4120].) Noted Microsoft behavior should be moved out into a separate document. Changed some second-person references in the introduction to identify the proper parties. Changed PA-CLIENT-CANONICALIZED to use a separate type for the actual referral data, add an extension marker to that type, and change the checksum key from the "returned session key" to the "AS reply key". Changed AD-LOGIN-ALIAS to contain a sequence of names, to be contained in AD-KDC-ISSUED instead of AD-IF-RELEVANT, and to drop the no longer needed separate checksum. Attempt to clarify the cache lifetime of referral information.

08 Moved Microsoft implementation info to appendix. Clarify lack of local server name canonicalization. Added optional authz-data for login alias, to support user-to-user case. Added requested-principal-name to ServerReferralData. Added discussion of caching information, and referral TGT lifetime.
07 Re-issued with new editor. Fixed up some references. Started history.

Authors’ Addresses

Kenneth Raeburn
Massachusetts Institute of Technology
77 Massachusetts Avenue
Cambridge, MA  02139
US

Email: raeburn@mit.edu
Larry Zhu
Microsoft Corporation
One Microsoft Way
Redmond, WA  98052
US

Email: lzhu@microsoft.com
This document is subject to the rights, licenses and restrictions contained in BCP 78, and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY, THE IETF TRUST AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at http://www.ietf.org/ipr.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Acknowledgment

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).