A Generalized PAC for Kerberos V5
draft-sorce-krbwg-general-pac-01

Abstract

This draft proposes a generalized authorization structure for the Kerberos V5 protocol. Such an authorization structure would allow for greater interoperability among directory services and other related Kerberos services across differing realms.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

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."

This Internet-Draft will expire on June 20, 2011.

Copyright Notice

Copyright (c) 2010 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as
Table of Contents

1. Introduction .............................................. 3
2. Requirements Language .................................... 3
3. Use-Case: Cross-Realm Directory Services .................. 3
   4.1. Attributes ............................................ 4
   4.2. PAD-Realm ............................................. 5
   4.3. PAD-Principal .......................................... 5
   4.4. PAD-Domain ............................................ 5
   4.5. PAD-Domain-UUID ....................................... 6
   4.6. PAD-Posix-Username .................................... 6
   4.7. PAD-Posix-UID ......................................... 6
   4.8. PAD-Posix-GID ......................................... 6
   4.9. PAD-Posix-Gecos ....................................... 6
   4.10. PAD-Posix-Homedir .................................... 6
   4.11. PAD-Posix-Shell ...................................... 6
   4.12. PAD-Fullname ......................................... 6
   4.13. PAD-AlternateNames .................................... 6
   4.14. PAD-Groups ........................................... 7
   4.15. PAD-MappedDomain-UUID ................................. 7
   4.16. PAD-Mapped-UID ....................................... 8
   4.17. PAD-Mapped-GID ....................................... 8
   4.18. PAD-Mapped-Attributes ................................. 8
   4.19. PAD-Mapped-Groups ................................... 8
5. Encoding ..................................................... 8
   5.1. PAD Format ............................................ 8
6. Data Structures and Extensions ................................ 10
   6.1. SignedPrincipalAuthorizationData ....................... 10
   6.2. GSS-API Authenticator Extension ....................... 11
7. Assigned numbers ........................................... 12
8. IANA Considerations ....................................... 12
9. Security Considerations .................................... 12
10. Acknowledgements ......................................... 12
11. References ............................................... 12
    11.1. Normative References ................................. 12
    11.2. Informative References .............................. 13
Appendix A. Additional Stuff .................................. 13
Authors’ Addresses .............................................. 13
1. Introduction

There is an increasing need today for Kerberos to support the delivery and processing of authorization information pertaining to the principals seeking access to the servers. Kerberos today is used extensively for authentication to directory services within the Enterprise. In many cases, a directory service is implemented as a distributed database system organized across multiple realms. As such, when a client in one realm seeks access to a directory service component located within a different realm, information regarding both the identity of the client and the permissions associated with that client must be communicated across the realms. Currently there does not exist a common and standardized structure in Kerberos (V5) for conveying access control or authorization information.

This draft proposes a general authorization structure for Kerberos that identifies a base set of common data elements or fields within the authorization structure, as well as the format of that structure. We refer to this data structure as the Principal Authorization Data (PAD) structure in order to distinguish it from existing structures, such as the Privilege Attribute Certificate defined by Microsoft in [MS-PAC].

2. Requirements Language

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 RFC 2119 [RFC2119].

3. Use-Case: Cross-Realm Directory Services

In this section we discuss one of the primary use-case scenarios for the Principal Authorization Data (PAD) structure within Kerberos V5. In this use-case a client principal is seeking to access a service in a different realm. Since the remote service does not have authorization information regarding the client, it needs to obtain it either from querying the directory service in its own realm or the directory service located in the client’s realm. It is here that a common PAD structure becomes necessary and invaluable in order to achieve a high-degree of interoperability between directory services in distinct realms.

In this use-case a client principal C1 in realm R1 is seeking access to services (or servers) located in a different realm R2. In accessing local service S1 in realm R1 the client must first be authenticated by KDC1 in that realm. A directory service (e.g.
LDAP) called D1 is used in realm R1 to perform authorization of the client, after the client has been authenticated by KDC1.

When the client principal later seeks to access services or resources S2 in realm R2, following the usual Kerberos flow the client must first obtain a cross-realm TGT from KDC1 (in realm R1) and then present it to KDC2 (in realm R2) in order to obtain a service-ticket for S2. However, one immediate issue is the fact that service S2 does not have authorization information regarding the permissions or privileges of client C1 in realm R1. The service S2 could query its own directory service D2 to obtain authorization information pertaining to client C1. In the absence of such information in D2, the service S2 could then perform a cross-realm query to the directory services D1 operating in realm R1.

However, this cross-realm query from S2 to D1 is not only inefficient, but it also implies knowledge of multiple heterogeneous systems by all actors. Two different realms may rely on completely different infrastructures for user information storage, ranging from different LDAP implementations with different schema conventions to NIS, SQL databases, flat files, and so on. Every service in the realm R2 would have to know what information system is in use in R1, how to reach it, how to read and eventually how to map data from it. Moreover security related aspects on the authentication of S2 by the directory D1, the authorization of S2 to make such a query, the protection of responses from D1 to S2, and so on, would have to be addressed.

This use-case illustrates the need for a common PAD structure to address this cross-realm authorization problem. In particular, the PAD structure for the cross-realm access to remote services needs to be contained or carried within cross-realm TGTs and service-tickets. Such a PAD structure needs to carry enough authorization information such that a decision can be made by service D2 in realm R2 regarding the access request originating from the client principal C1 within realm R1.


4.1. Attributes

The following attributes are defined in this document:

- PAD-Realm
- PAD-Principal
4.2.  PAD-Realm

The full Realm Name of the Realm the principal belongs to.

4.3.  PAD-Principal

The name of the principal.  Joined with the PAD-Realm component it
MUST match the full principal name of the owner of the ticket.

4.4.  PAD-Domain

A short domain name that uniquely identifies, within the set of
trusted realms, the domain the principal belongs to.  The short
Domain name is useful for representation purposes in the OS.
4.5. PAD-Domain-UUID

A UUID that universally identifies the domain the following local
identifiers belongs to. This is used to differentiate between local
identifiers belonging to different domains/realms.

4.6. PAD-Posix-Username

This is the user name that correspond to the kerberos principal, this
is the name that SHOULD be used by the OS to represent the user. The
OS may decide to prefix or posfix this name with the PAD-Domain or
PAD-Realm names in case of name conflicts with local accounts.

4.7. PAD-Posix-UID

This is the UID Number associated to the user. This number is local
to the domain identified by PAD-Domain-UUID.

4.8. PAD-Posix-GID

This is the Primary GID Number associated to the user. This number
is local to the domain identified by PAD-Domain-UUID.

4.9. PAD-Posix-Gecos

The Gecos field for the User associated to the Principal if
available. Can be omitted. If not available PAD-Fullname can be
used instead.

4.10. PAD-Posix-Homedir

The home directory path relative to the local system, if available.
If not available local defined defaults apply.

4.11. PAD-Posix-Shell

The default shell for the user, defined as the path of the binary
relative to the local filesystem, if available. If not available
local defined defaults apply.

4.12. PAD-Fullname

The full name of the user if available.

4.13. PAD-AlternateNames

Alternate names can be used by application to identify a user by
means that differ from the user principal. Names are in string form.
and utf8 encoded [UTF-8]. In order to allow applications to recognize the name type without guesswork, alternate names are prefixed with a string followed by the colon ‘:’ character and the name, without any space or other separation character. The following alternate names are currently recognized: EMAIL, OS, OPENID, OAUTH It is allowed to include multiple alternate names of the same type. The order in which they are provided represent the priority within the same name type, if applications need to choose between names.

(TODO: need discussion on whether these needs labeled prefixes or explicit attributes for each alternate representation etc...)

4.14. PAD-Groups

This is a structured attribute and defines the groups the principal is member of.

The first value in the structure represents the Domain-UUID and is optional. If missing the Domain UUID is assumed to be the one defined in the PAD-Domain-UUID attribute.

Then an array of values that define the groups as follows. Each group value includes 2 subvalues:

- (1) GID: This is the gid number of the group.
- (2) Name: This is the name of the group.

Multiple PAD-Groups attributes can be present at the same time. A trusting KDC can augment the original user’s set of groups by adding a new PAD-Groups structure that contains groups local to the trusting domain. In this case the Domain-UUID is required. The Domain UUID is used for gid number conflict resolution when the PAC is transmitted between services of different realms.

PAD-Groups are optional attributes and the KDC, upon PAC revalidation, may decide to remove the original attributes that do not belong to the KDC security domain in order to save space or to censor information to avoid disclosing data to services.

4.15. PAD-MappedDomain-UUID

UUID of the domain local identifiers belong to, after mapping from the original domain. (NB. Needs better explanation and discussion)
4.16. PAD-Mapped-UID

The value to be used instead of PAD-Posix-UID, and local to PAD-MappedDomain-UUID.

4.17. PAD-Mapped-GID

The value to be used instead of PAD-Posix-GID, and local to PAD-MappedDomain-UUID.

4.18. PAD-Mapped-Attributes

In POSIX, users and groups ID are not universally unique, and different Realms (even different machines within an authorization realm actually) may have overlapping and conflicting IDs. If this is the case, a trusting KDC may decide to re-map IDs coming from a foreign Realm to help services with uid/gid mapping and avoid ID conflicts that can lead to serious security issues. The original IDs are generally preserved.

If PAD-MappedDomain-UUID is recognized by the application to be the local security domain identifier, then only the mapped attributes SHOULD be used for authorization purposes and the original values SHOULD be ignored.

4.19. PAD-Mapped-Groups

The mapped groups to be used instead of PAD-Groups, the structure definition is the same as that of PAD-Groups.

5. Encoding

The Kerberos protocol is defined in [RFC4120] using Abstract Syntax Notation One (ASN.1) [X680]. As such, this specification also uses the ASN.1 syntax for specifying both the abstract layout of the PAD attributes, as well as their encodings.

5.1. PAD Format

The information carried in the PAD need to be augment by some control information and packaged in a way that makes it possible to devise future extensions.

Additional information needed to validate the PAD:

- The expiration time (must be the same as the ticket expiration time).
- The principal name (must be the same principal that owns the ticket).

- The KDC signature (for re-validation purposes).

- The Service Signature (in order to trust the PAD has not been tampered with).

This information is needed to validate the PAD and make sure it is not modified, outdated, or information belonging to a different principal.

In order to make the PAD extensible and at the same time always verifiable we propose that the PAD is embedded in a ASN.1 structure that can contain multiple optional buffers identified by numbers (how to assign numbers TBD).

Buffer number 0 is an ASN.1 structure that includes all attributes described in paragraph 4. This buffer is itself optional.

The whole structure with all its buffers is what is signed with the KDC and the service keys.

The final structure to be included in AD-IF-RELEVANT container and looks loosely like the following diagram.

```
<table>
<thead>
<tr>
<th>PAD:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service Signature (Checksum)</td>
</tr>
<tr>
<td>----------------------------------</td>
</tr>
<tr>
<td>KDC Signature (Checksum)</td>
</tr>
<tr>
<td>----------------------------------</td>
</tr>
<tr>
<td>-PAD-DATA:------------------------</td>
</tr>
<tr>
<td>principal name</td>
</tr>
<tr>
<td>ticket expiration time</td>
</tr>
<tr>
<td>Buf 0: --- (optional)---------</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>---------------------------</td>
</tr>
<tr>
<td>...</td>
</tr>
<tr>
<td>Buf X: --- (optional)------</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>---------------------------</td>
</tr>
</tbody>
</table>
```
6. Data Structures and Extensions

6.1. SignedPrincipalAuthorizationData

AD-SIGNED-PAD ::= SEQUENCE {
  kdc-signature [0] Checksum,
  svc-signature [1] Checksum,
}

kdc-signature
A cryptographic checksum computed over the encoding of the
pad-data field, keyed with the krbtgt key.
Checksum type TBD.

svc-signature
A cryptographic checksum computed over the encoding of the
pad-data field, keyed with the service long term key.
Checksum type TBD.

PAD-DATA ::= SEQUENCE {
  p-realm  [0] Realm,
  p-name   [1] PrincipalName,
}

p-realm, p-name
The realm and name of the principal the authorization data
elements apply to.

elements
A sequence of authorization data elements issued by the KDC.

The AD-SIGNED-PAD data is intended to provide a means for a Kerberos
principal credentials to carry authorization data that the receiving
service can use to perform authorization operations.

The KDC signature is required to allow the KDC to validate the data without having to recompute the contents at every TGS request.

The SVC signature is required so that the Service can verify that the authorization data has been validated by the KDC.

The ad-type for AD-SIGNED-PAD is (X).

6.2. GSS-API Authenticator Extension

The Authenticator Checksum as defined in RFC 4121 limit the size of delegated credentials in the KRB_CRED message to a size of 64KiB.

In order to be able to transfer larger messages an extension is defined. This extension is used in stead of the Dlght/Deleg fields, and the Dlght and Deleg fields MUST not be included when this extensions is appended to the authenticator.

The extension SHALL have the following format:

<table>
<thead>
<tr>
<th>Octet</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0..1</td>
<td>ExtN</td>
<td>A 16bit value identifying the extension.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Represented in little-endian order;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Contains the hex value XX XX (XXXX).</td>
</tr>
<tr>
<td>2..5</td>
<td>Length</td>
<td>The length of the Extended Delegation field.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Represented in little-endian order;</td>
</tr>
<tr>
<td>6..N</td>
<td>Data</td>
<td>A KRB_CRED message (N = Length + 6)</td>
</tr>
</tbody>
</table>

A new flag GSS_C_EXT_DELEG_FLAG with Value X is also defined. This flag is used instead of GSS_C_DELEG_FLAG when the delegated credentials are larger than 64KiB and cannot fit in the standard Deleg field.

Implementors SHOULD use this Extensions and this flag only if the KRB_CRED message is larger than 64KiB and use the standard Deleg field otherwise.
7. Assigned numbers

TBD

8. IANA Considerations

TBD.

9. Security Considerations

Although it is anticipated that the PAD structure itself will be carried within a ticket and thereby protected using the existing encryption methods on that ticket, there are a number of issues that have bearings on the security of the entire Kerberos realm as a whole. Some of these issues are as follows:

- **UID and GUID Collisions**: There is always the possibility of collision of numbers repressing a UID and a GID. This problem can be remedied to a large degree by realm using an appropriate number selection policy and algorithms.

- **Transit-domain issues**: The PAC must be signed by the KDC that is attaching it to a ticket with 2 different signatures. The service signature so that the service can verify its KDC validated the contents. The KDC signature, so that the OS can ask the KDC to confirm the PAD has not been modified by a less trusted service. For cross-realm tickets the "service" signature is made with the cross-realm key. When a KDC receives a PAD it is allowed to modify it in any way. It can filter out information or add information (like group memberships defined locally). A KDC may also decide to change information in different ways depending on what service it is targeted to.

10. Acknowledgements

TBD.

11. References

11.1. Normative References

11.2. Informative References

[MIT-Athena]

[MS-PAC]

[POSIX]

[RFC1510]

[RFC2119]
Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2307]

[RFC3552]

[X.690]

Appendix A. Additional Stuff

This becomes an Appendix.
Authors’ Addresses

Simo Sorce
Red Hat

Email: ssorce@redhat.com

Tom Yu
MIT Kerberos Consortium

Email: tlyu@mit.edu

Thomas Hardjono
MIT Kerberos Consortium

Email: hardjono@mit.edu