Network Working Group
Request for Comments: 5021
Updates: 4120
Category: Standards Track

S. Josefsson
SJD

August 2007

Extended Kerberos Version 5 Key Distribution Center (KDC) Exchanges over TCP

Status of This Memo

Copyright Notice

Abstract

This document describes an extensibility mechanism for the Kerberos V5 protocol when used over TCP transports. The mechanism uses the reserved high-bit in the length field. It can be used to negotiate TCP-specific Kerberos extensions.

Table of Contents

1. Introduction

The Kerberos V5 [3] specification, in section 7.2.2, reserves the high order bit in the initial length field for TCP transport for future expansion. This document updates [3] to describe the behaviour when that bit is set. This mechanism is intended for extensions that are specific for the TCP transport.

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 RFC 2119 [1].

3. Extension Mechanism for TCP Transport

The reserved high bit of the request length field is used to signal the use of this extension mechanism. When the reserved high bit is set in the length field, the remaining 31 bits of the initial 4 octets are interpreted as a bitmap. Each bit in the bitmask can be used to request a particular extension. The 31 bits form the "extension bitmask". It is expected that other documents will describe the details associated with particular bits.

A 4-octet value with only the high bit set, and thus the extension bitmask all zeros, is called a PROBE. A client may send a probe to find out which extensions a KDC supports. A client may also set particular bits in the extension bitmask directly, if it does not need to query the KDC for available extensions before deciding which extension to request.

Note that clients are not forced to use this extension mechanism, and further, clients are expected to only use it when they wish to negotiate a particular extension.

The protocol is as follows. The client MUST begin by sending a 4-octet value with the high bit set. The packet is thus either a PROBE or a specific request for some extension(s). The client MUST NOT send additional data before the server has responded.

If a KDC receives a request for a set of extensions that it supports, it MUST respond by sending a 4-octet zero value, i.e., 0x00000000. The KDC MAY directly send additional data after the zero value, without waiting for the client to respond, as specified by the particular negotiated extension. (Note: A 4-octet zero value can never be sent by an implementation that conforms to RFC 4120 and that does not support this extension mechanism, because a KRB-ERROR is always of non-zero size.)

If a KDC receives a PROBE, or if a KDC does not support all extensions corresponding to set bits in the extension bitmask, the KDC MUST return 4 octets with the high bit set, and with the remaining bitmask indicating which extensions it supports. The KDC then MUST wait, and the client MUST send a second 4-octet value with the high bit set. If the second 4-octet value is a PROBE or an unsupported extension, the KDC MUST close the connection. This can be used by the client to shut down a session when the KDC did not support an extension that is required by the client. If the second 4-octet value is a supported extension, the KDC MUST respond by sending a 4-octet zero value, i.e., 0x00000000. The KDC MAY directly send additional data after the zero value, as specified by the particular negotiated extension.

The client and KDC SHOULD wait for the other side to respond according to this protocol, and the client and KDC SHOULD NOT close the connection prematurely. Resource availability considerations may influence whether, and for how long, the client and KDC will wait for the other side to respond to a request.

The KDC MUST NOT support the extension mechanism if it does not support any extensions. If no extensions are supported, the KDC MUST return a KRB-ERROR message with the error KRB_ERR_FIELD_TOOLONG and MUST close the TCP stream, similar to what an implementation that does not understand this extension mechanism would do.

The behaviour when more than one non-high bit is set depends on the particular extension mechanisms. If a requested extension (bit X) does not specify how it interacts with another requested extension (bit Y), the KDC MUST treat the request as a PROBE or unsupported extension, and proceed as above.

Each extension MUST describe the structure of protocol data beyond the length field, and the behaviour of the client and KDC. In particular, the structure may be a protocol with its own message framing. If an extension mechanism reserves multiple bits, it MUST describe how they interact.

4. Interoperability Consideration

Implementations with support for TCP that do not claim to conform to RFC 4120 may not handle the high bit correctly. The KDC behaviour may include closing the TCP connection without any response, and logging an error message in the KDC log. When this was written, this problem existed in modern versions of popular KDC implementations. Implementations experiencing trouble getting the expected responses from a KDC might assume that the KDC does not support this extension mechanism. A client might remember this semi-permanently, to avoid

triggering the same problematic behaviour on the KDC every time. Care should be taken to avoid unexpected behaviour for the user when the KDC is eventually upgraded. Implementations might also provide a way to enable and disable this extension on a per-realm basis. How to handle these backwards compatibility quirks are in general left unspecified.

5. Security Considerations

Because the initial length field is not protected, it is possible for an active attacker (i.e., one that is able to modify traffic between the client and the KDC) to make it appear to the client that the server does not support this extension mechanism (a downgrade attack). Further, active attackers can also interfere with the negotiation of which extensions are supported, which may also result in a downgrade attack. This problem can be solved by having a policy in the clients and in the KDC to reject connections that do not have the desired properties. The problem can also be mitigated by having the negotiated extension send a cryptographic checksum of the offered extensions.

6. IANA Considerations

IANA has created a new registry for "Kerberos TCP Extensions". The initial contents of this registry are:

Bit #		Reference
0..29	AVAILABLE for registration.	
30	RESERVED.	RFC 5021

IANA will register values 0 to 29 after IESG Approval, as defined in BCP 64 [2]. Assigning value 30 requires a Standards Action that updates or obsoletes this document.

Registration policy: The IESG will act as a steward for the namespace, considering whether the registration is justified given the limited size of the namespace. The IESG will also confirm that proposed registrations are not harmful to the Internet.

7. Acknowledgements

Nicolas Williams, Jeffrey Hutzelman, Sam Hartman, and Chris Newman provided comments that improved the protocol and document.

Thanks to Andrew Bartlett who pointed out that some implementations (MIT Kerberos and Heimdal) did not follow RFC 4120 properly with regards to the high bit, which resulted in an Interoperability Consideration.

8. Normative References

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

[2] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 2434, October 1998.

[3] Neuman, C., Yu, T., Hartman, S., and K. Raeburn, "The Kerberos Network Authentication Service (V5)", RFC 4120, July 2005.

Appendix A. Copying Conditions

Author's Address

Simon Josefsson
SJD

EMail: simon@josefsson.org

Full Copyright Statement

Intellectual Property

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.