Using Kerberos V5 over the Transport Layer Security (TLS) protocol
draft-josefsson-kerberos5-starttls-09

Abstract

This document specify how the Kerberos V5 protocol can be transported over the Transport Layer Security (TLS) protocol, to provide additional security features.

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 February 15, 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 described in the Simplified BSD License.

This document may contain material from IETF Documents or IETF
Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

Table of Contents

1. Introduction and Background .................................. 3
2. Kerberos V5 STARTTLS Extension .............................. 4
3. Examples ....................................................... 5
4. STARTTLS aware KDC Discovery ................................. 6
5. Server Certificates ............................................ 7
6. IANA Considerations ........................................... 9
7. Acknowledgements .............................................. 10
8. Security Considerations ....................................... 11
9. References ..................................................... 12
   9.1. Normative References .................................... 12
   9.2. Informative References .................................. 12
8. References ..................................................... 12
Author’s Address .................................................. 13
1. Introduction and Background

This document describe how a Kerberos V5 [RFC4120] implementation may upgrade communication between clients and Key Distribution Centers (KDCs) to use the Transport Layer Security (TLS) [RFC5246] protocol.

The TLS protocol offer integrity and privacy protected exchanges that can be authentication using X.509 certificates, OpenPGP keys [RFC5081], and user name and passwords via Secure Remote Password (SRP) [RFC5054].

There are several reasons to use Kerberos V5 over TLS.

- Prevents downgrade attacks affecting, e.g., encryption types and pre-auth data negotiation. The encryption type field in KDC-REQ, and the METHOD-DATA field with the requested pre-auth types from the server in KDC_ERR_PREAMTH_REQUIRED errors in KDC-REP, are sent without integrity or privacy protection in Kerberos 5. This allows an active attacker to replace the encryption type with a compromised encryption type, e.g., 56-bit DES, or request that clients should use a broken pre-auth type. Since clients in general cannot know the encryption types other servers support, or the pre-auth types servers prefer or require, it is difficult for the client to detect if there was a man-in-the-middle or if the remote server simply did not support a stronger encryption type or preferred another pre-auth type.

- Kerberos exchanges are privacy protected. Part of many Kerberos packets are transferred without privacy protection (i.e., encryption). That part contains information, such as the client principal name, the server principal name, the encryption types supported by the client, the lifetime of tickets, etc. Revealing such information is, in some threat models, considered a problem.

- Additional authentication against the KDC. In some situations, users are equipped with smart cards with a RSA authentication key. In others, users have a OpenPGP client on their desktop, with a public OpenPGP key known to the server.

- Explicit server authentication of the KDC to the client. In traditional Kerberos 5, authentication of the KDC is proved as a side effect that the KDC knows your encryption key (i.e., your password).

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].
2. Kerberos V5 STARTTLS Extension

The STARTTLS extension uses the Kerberos V5 TCP extension mechanism [RFC5021]. The extension uses bit #TBD in the extension bitmask.

The protocol is as follows. The client requests the extension by setting the STARTTLS bit in the TCP extension mechanism bitmask. (How to deal with extension negotiation failures at this point is described in [RFC5021].) After the server has sent the 4-octet value 0x00000000 to indicate support of this extension, the stream will be controlled by the TLS protocol and its framing. The TLS protocol is initiated by the client.

Typically, the client initiate the TLS handshake protocol by sending a client hello, and the server responds, and the handshake continues until it either succeed or fails.

If for any reason the handshake fails, the STARTTLS protocol will also fail, and the TLS error is used as the error indication. In this case, no further messages can be exchanged over the same TCP session.

If the handshake succeeds, the Kerberos V5 authentication protocol is performed within the protected TLS channel, like a normal TCP Kerberos V5 exchange. In particular, this means that every Kerberos V5 packet will be prefixed by a 4-octet length field, that indicate the length of the Kerberos V5 packet.

When no further Kerberos V5 messages needs to be transferred in the TLS session, the TLS session MUST be shut down properly using the close_notify alert. When the TLS session is shut down, the TCP connection cannot be re-used to send any further data and MUST be closed.
3. Examples

A complete packet flow for a successful AS-REQ/REP exchange protected by this mechanism will be as follows. The "STARTTLS-bit" is a 4-octet value with only the bit allocated for this extension set, and | is the binary OR operation.

Client                                               Server

[ Kerberos V5 TCP extension mechanism negotiation starts ]

0x80000000 | STARTTLS-bit   -------->  0x00000000
            <--------

[ TLS negotiation starts ]

ClientHello            -------->  ServerHello
Certificate*            Certificate*
ServerKeyExchange*      ServerKeyExchange*
CertificateRequest*     CertificateRequest*
                        <--------  ServerHelloDone

Certificate*
ClientKeyExchange
CertificateVerify*
[ChangeCipherSpec]
Finished                -------->  [ChangeCipherSpec]
                          <--------  Finished

[ Kerberos V5 negotiation starts ]

4 octet length field
Kerberos V5 AS-REQ       -------->  4 octet length field
                               Kerberos V5 AS-REP
                               <--------

* Indicates optional or situation-dependent messages that are not always sent.
4. STARTTLS aware KDC Discovery

Section 7.2.3 of Kerberos V5 [RFC4120] describe how Domain Name System (DNS) SRV records [RFC2782] can be used to find the address of an KDC. We define a new Service of "kerberos-tls" to indicate that the particular KDC is intended to support this STARTTLS extension. The Proto (tcp), Realm, TTL, Class, SRV, Priority, Weight, Port and Target have the same meaning as in RFC 4120.

For example:

```plaintext
_kerberos-tls._tcp.EXAMPLE.COM. IN SRV 0 0 88 kdc1.example.com.
_kerberos-tls._tcp.EXAMPLE.COM. IN SRV 1 0 88 kdc2.example.com.
```
5. Server Certificates

The TLS protocol may be used in a mode that provides server authentication using, for example, X.509 and OpenPGP.

A goal for the protocol described in this memo is that it should be as easy to implement and deploy on clients as support for UDP/TCP. Since many client environments do not have access to long-term storage, or to long-term storage that is sufficiently secure to enable validation of server certificates, the Kerberos V5 STARTTLS protocol does not require clients to verify server certificates. If server certification had been required, then environments with constrained clients such as those mentioned would be forced to disable TLS; this would arguably be worse than TLS without server certificate validation as use of TLS, even without server certificate validation, protects against some attacks that Kerberos V5 over UDP/TCP do not. For example, even without server certificate validation, TLS does protect against passive network sniffing aimed at tracking Kerberos service usage by a given client.

Note however that use of TLS without server certificate verification opens up for a range of active attacks such as man-in-the-middle.

When clients have the ability, they MUST validate the server certificate. For this reason, if a KDC presents a X.509 server certificate over TLS, it MUST contain an otherName Subject Alternative Name (SAN) identified using a type-id of id-krb5starttls-san. The intention is to bind the server certificate to the Kerberos realm for the purpose of using Kerberos V5 STARTTLS. The value field of the otherName should contain the realm as the "Realm" ASN.1 type.

id-krb5starttls-san OBJECT IDENTIFIER ::= 
    { iso(1) identified-organization(3) dod(6) internet(1) 
    private(4) enterprise(1) gnu(11591) 
    shishi(6) krb5starttls-san(1) }

To validate a server certificate, the client MAY use local configuration (e.g., a list that maps the Kerberos realm to a copy of the server’s certificate) and compare that with the authentication information provided from the server via TLS. For illustration, the server certificate could be a X.509 certificate or an OpenPGP key. In this mode, the client need no processing related to id-krb5starttls-san.

When the server presents a X.509 server certificate, clients MAY use "Certification Path Validation" as described in [RFC5280] to validate the KDC server certificate. In addition, unless the client can otherwise verify that the server certificate is bound to the KDC of
the target realm, the client MUST verify that the server certificate contains the id-krb5starttls-san SAN and that the value is identical to the intended Kerberos realm.
6. IANA Considerations

The IANA is requested to allocate a bit in the "Kerberos TCP Extensions" registry for the extension described in this document, as per [RFC5021].
7. Acknowledgements

Miguel A. Garcia, Jeffrey Hutzelman, Sam Hartman, Magnus Nystroem, and Peter Saint-Andre (in alphabetical order) provided comments that improved the protocol and document.
8. Security Considerations

The security considerations in Kerberos V5, TLS, and the Kerberos V5 TCP extension mechanism are inherited.

Note that TLS does not protect against Man-In-The-Middle (MITM) attacks unless clients verify the KDC’s credentials (X.509 certificate, OpenPGP key, etc) correctly. Although certificate validation adds an extra layer of protection, that is not considered strictly necessary to improve the security profile of Kerberos V5 as outlined in this document.

If server authentication is used, some information about the server (such as its name) is visible to passive attackers.

To protect against the inherent downgrade attack in the extension framework, implementations SHOULD offer a policy mode that requires this extension to always be successfully negotiated, for a particular realm, or generally. For interoperability with implementations that do not support this extension, the policy mode SHOULD be disabled by default.
9. References

9.1. Normative References


9.2. Informative References

