                       Using Kerberos Version 5
              over the Transport Layer Security (TLS) Protocol

Abstract

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

Status of This Memo

   This document is not an Internet Standards Track specification; it is
   published for informational purposes.

   This document is a product of the Internet Engineering Task Force
   (IETF). It represents the consensus of the IETF community. It has
   received public review and has been approved for publication by the
   Internet Engineering Steering Group (IESG). Not all documents
   approved by the IESG are a candidate for any level of Internet
   Standard; see Section 2 of RFC 5741.

   Information about the current status of this document, any errata,
   and how to provide feedback on it may be obtained at
   http://www.rfc-editor.org/info/rfc6251.

Table of Contents

1. Introduction and Background

   This document describes 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 offers integrity- and privacy-protected exchanges
   that can be authenticated using X.509 certificates, OpenPGP keys
   [RFC6091], and usernames and passwords via Secure Remote Password
   (SRP) [RFC5054].

   There are several reasons to use Kerberos V5 over TLS.

   o It 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_PREAUTH_REQUIRED errors in
      KDC-REP are sent without integrity or privacy protection in
      Kerberos V5. 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.

o Kerberos exchanges are privacy protected. Parts 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.

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

o It provides explicit server authentication of the KDC to the
client. In traditional Kerberos V5, 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 0 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 initiates the TLS handshake protocol by sending
a client hello, the server responds, and the handshake continues
until it either succeeds 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 indicates
the length of the Kerberos V5 packet.

When no further Kerberos V5 messages need 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*
                                                 ServerKeyExchange*
                                                 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] describes how Domain Name
   System (DNS) SRV records [RFC2782] can be used to find the address of
   a 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:

   _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 the use of TLS, even without server
   certificate validation, protects against some attacks that Kerberos
   V5 over UDP/TCP does 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.

   However, note that the use of TLS without server certificate
   verification opens up 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 an 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 an X.509 certificate or an OpenPGP key.
   In this mode, the client needs no processing related to id-
   krb5starttls-san.

   When the server presents an 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

   Per [RFC5021], the IANA has allocated a bit (value 0) in the
   "Kerberos TCP Extensions" registry for Krb5 over TLS, the extension
   described in this document.

7. Acknowledgements

   Miguel A. Garcia, Sam Hartman, Jeffrey Hutzelman, 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 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

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

   [RFC2782] Gulbrandsen, A., Vixie, P., and L. Esibov, "A DNS RR for
              specifying the location of services (DNS SRV)", RFC 2782,
              February 2000.

   [RFC4120] Neuman, C., Yu, T., Hartman, S., and K. Raeburn, "The
              Kerberos Network Authentication Service (V5)", RFC 4120,
              July 2005.

   [RFC5021] Josefsson, S., "Extended Kerberos Version 5 Key
              Distribution Center (KDC) Exchanges over TCP", RFC 5021,
              August 2007.

   [RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security
              (TLS) Protocol Version 1.2", RFC 5246, August 2008.

   [RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S.,
              Housley, R., and W. Polk, "Internet X.509 Public Key
              Infrastructure Certificate and Certificate Revocation List
              (CRL) Profile", RFC 5280, May 2008.

9.2. Informative References

   [RFC5054] Taylor, D., Wu, T., Mavrogiannopoulos, N., and T. Perrin,
              "Using the Secure Remote Password (SRP) Protocol for TLS
              Authentication", RFC 5054, November 2007.

   [RFC6091] Mavrogiannopoulos, N. and D. Gillmor, "Using OpenPGP Keys
              for Transport Layer Security (TLS) Authentication",
              RFC 6091, February 2011.

Author's Address

   Simon Josefsson
   Simon Josefsson Datakonsult AB
   Hagagatan 24
   Stockholm 113 47
   Sweden

   EMail: simon@josefsson.org
   URI: http://josefsson.org/