Strengthening the Kerberos V5 Reply Key using TLS
draft-josefsson-krb5starttls-bootstrap-00

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and 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 September 3, 2009.

Copyright Notice

Copyright (c) 2009 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 in effect on the date of publication of this document (http://trustee.ietf.org/license-info). Please review these documents carefully, as they describe your rights and restrictions with respect to this document.
Abstract

This document describes how to strengthen the Kerberos V5 reply key using keying material derived from TLS, by using a pre-authentication mechanism. The goals are to 1) allow clients to securely learn a realm’s KDC X.509 certificate, 2) distribute the X.509 trust anchors used by the KDC, and 3) make it possible for clients to use Kerberos V5 over TLS without having to validate the server certificates.

Table of Contents

1. Introduction and Background .......................... 3
2. Protocol ............................................. 4
3. IANA Considerations .................................. 6
4. Acknowledgements .................................... 7
5. Security Considerations ............................... 8
6. References ........................................... 9
   6.1. Normative References .............................. 9
   6.2. Informative References ............................ 9
Author’s Address ...................................... 10
1. Introduction and Background

This document describes a Kerberos V5 [RFC4120] pre-authentication mechanism that uses Kerberos V5 over TLS [I-D.josefsson-kerberos5-starttls] to achieve:

- Allow Kerberos V5 clients to securely learn a Kerberos V5 realm’s Key Distribution Center (KDC) certificates. This is achieved by having the client connect to a KDC, take a note of the server’s certificate, and verify them as belonging to the KDC the user trusts by properly decrypting the Kerberos V5 response using the user’s password. Only the correct KDC will be able to generate a Kerberos V5 response using the user’s password and the secrets derived from the TLS channel.

- Securely distribute the trust anchors used by the Key Distribution Center (KDC) in a Kerberos V5 realm. This is achieved the same way as before, but rather than remembering the server certificate, it remembers the trust anchor.

- The ability to use Kerberos V5 over TLS without having to validate the server certificates.

The mechanism to achieve the above goals is for the KDC to strengthen the Kerberos V5 reply key using keying material derived from the TLS channel [RFC5246] using the algorithm specified in Keying Material Exporters for Transport Layer Security (TLS) [I-D.ietf-tls-extractor].

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. Protocol

The client and KDC MUST support Kerberos V5 over TLS [I-D.josefsson-kerberos5-starttls]. If the client do not (yet) have trust anchors for the KDC, it should disable verification of the server certificate.

To signal that the client wishes the KDC to strengthen the reply key using keying material derived from the TLS session, it sends a pre-authentication mechanism called "pa-krb5starttls-bootstrap". It has a pdata-type integer value of #TBD.

The pre-authentication structure is defined in RFC 4120 as:

```
PA-DATA ::= SEQUENCE {
    padata-type     [1] Int32,
    padata-value    [2] OCTET STRING -- might be encoded AP-REQ
}
```

The content of the padata-value should be the DER encoding of the empty string.

When receiving the request to use the "pa-krb5starttls-bootstrap" pre-authentication message, the KDC needs to decide whether to honor it or not. This is a policy decision that can depend on several reasons, including the content of the request. If the KDC decides that it does not wish to honor the "pa-krb5starttls-bootstrap" request, the KDC MUST fail the request by returning KDC_ERR_PREAUTH_FAILED.

When the KDC decides to honor the client’s request, it will process the incoming request as usual except that the KDC-REP reply key is post processed as follows. The post processing uses Keying Material Exporters for Transport Layer Security (TLS) [I-D.ietf-tls-extractor]. The channel binding input "tlscb" value MUST be the client’s TLS Finished message data as described in the "tls-unique" channel binding registration.
StrengthenKrb5ReplyKeyUsingTLS (inkey, inkey_len, 

tlscb, tlscb_len)

Input:          inkey      encryption key, an octet string
inkey_len  length of encryption key, 
a positive integer 

tlscb      channel binding data, an octet string, 

tlscb_len  length of channel binding data, 
a positive integer

Output:         outkey   derived key, a inlen-octet string

Steps:

1. Perform the TLS Exporter step:

   outkey = PRF(master_secret, label, 
     SecurityParameters.client_random + 
     SecurityParameters.server_random + 
     context_value_length + context_value 
   )[length]

   The "context_value" should be the concatenation of "inkey" 
   followed by "tlscb".

   Consequently, the length of "context_value" (which used to 
   derived "context_value_length") will be the sum of 
   "inkey_len" and "tlscb_len".

   Use the value of "inkey_len" as the value of the "length" 
   variable.

3. Output the derived key "outkey".

The client will strengthen the KDC-REP reply key using the same 
procedure.

On successful decryption of the KDC-REP, the clients is certain that 
it is talking to a KDC that knows the client’s shared key without any 
man-in-the-middle. The client can then remember the KDC server 
certificate and/or trust anchors transferred during the TLS 
handshake, to be used during future Kerberos V5 over TLS connections. 
The client MAY skip using this protocol for future connections, and 
instead rely on the standard Kerberos V5 over TLS protocol with 
proper validation of server certificate.
3. IANA Considerations

None.
4. Acknowledgements

Nicolas Williams mentioned the advantages in
5. Security Considerations

The security considerations in Kerberos V5 [RFC4120], TLS [RFC5246], Kerberos V5 TCP extension [RFC5021], and Kerberos V5 over TLS [I-D.josefsson-kerberos5-starttls] are inherited.
6. References

6.1. Normative References


6.2. Informative References


Author’s Address

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

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