Internet X.509 Public Key Infrastructure -- HTTP Transport for CMP
draft-ietf-pkix-cmp-transport-protocols-14.txt

Abstract

   This document describes how to layer the Certificate Management
   Protocol over HTTP. It is the "CMPtrans" document referenced in RFC
   4210 and therefore updates the reference given therein.

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 April 1, 2012.

Copyright Notice

Table of Contents

[1](). Introduction

   The Certificate Management Protocol (CMP) [[RFC4210]()] requires a well
   defined transport mechanism to enable End Entities (EEs),
   Registration Authorities (RAs) and Certificate Authorities (CAs) to
   pass PKIMessage sequences between them. This document defines the
   transport mechanism which was removed from the main CMP specification
   with the second release and referred to be in a separate document.

   The first version of the CMP specification [[RFC2510]()] included a brief
   description of a simple transport protocol layer on top of TCP. Its
   features was simple transport level error-handling and a mechanism to
   poll for outstanding PKI messages. Additionally it was mentioned
   that PKI messages could also be conveyed using file-, E-mail- and
   HTTP-based transport, but those were not specified in detail.

   The current version of the CMP specification [[RFC4210]()] incorporated
   an own polling mechanism and thus the need for a transport protocol
   providing this functionality vanished. The remaining features CMP
   requires from its transport protocols are connection and error
   handling.

   During the long time it existed as draft, this RFC was undergoing
   drastic changes. The "TCP-Based Management Protocol" was enhanced
   and a TCP-Messages-over-HTTP transport specification appeared. As
   both proved to be needless and cumbersome, implementers preferred to
   use plain HTTP transport. This document now reflects that by
   exclusively describing HTTP as transport protocol for CMP.

   HTTP transport is generally easy to implement, traverses network
   borders utilizing ubiquitous proxies and is already commonly found in
   existing implementations.

   With its status codes HTTP provides needed error reporting
   capabilities. General problems on the server side as well as those
   directly caused by the respective request can be reported to the
   client.

   As CMP implements a transaction ID, identifying transactions spanning
   over more than just a single request/response pair, the statelessness
   of HTTP is not blocking its usage as transport protocol for CMP
   messages.

2. Requirements

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 [RFC2119].

3. HTTP-Based Protocol

For direct interaction between two entities, where a reliable
transport protocol like TCP is available, HTTP SHOULD be utilized for
conveying CMP messages.

3.1. HTTP Versions

Either HTTP/1.0 as described in [RFC1945] or HTTP/1.1 as in [RFC2616]
MAY be used. Server implementations SHOULD be able to interact with
counterparts utilizing either HTTP protocol version.

3.2. Persistent Connections

HTTP permits to reuse a connection for subsequent requests.
Implementations may use this functionality for messages within the
same transaction but MUST NOT rely on that as e.g. intermediate HTTP
proxies might terminate the connection after each request/response
pair.

3.3. General Form

An ASN.1 DER-encoded PKIMessage is sent as the entity-body of an HTTP
POST request. If this HTTP request is successful, the server returns
the CMP reply in the body of the HTTP response. The response status
code in this case MUST be 200; other 2xx codes MUST NOT be used for
this purpose. HTTP responses to pushed CMP Announcement messages
utilize the status codes 201 and 202 to identify whether the received
information was processed.

All applicable 4xx or 5xx status codes may be used to inform the
client about errors.

3.4. Media Type

The Internet Media Type "application/pkixcmp" MUST be set in the HTTP
header when conveying a PKIMessage.

3.5. Communication Workflow

In CMP most communication is initiated by the end entities where
every CMP request triggers a CMP response message from the CA or RA.

The CMP Announcement messages described in Section 3.7 are an
exception. Their creation may be triggered by certain events or done
on a regular basis by a CA. The recipient of the Announcement only
replies with an HTTP status code acknowledging the receipt or
indicating an error but not with a CMP response.

If the receipt of an HTTP request is not confirmed by receiving an
HTTP response, it MUST be assumed that the transported CMP message
was not successfully delivered to its destination.

3.6. HTTP Request-URI

The Request-URI is formed as specified in [RFC3986].

Client requests containing a PKI message MUST be directed to an
Request-URI depicting a directory. A server implementation MUST
handle Request-URIs with or without a trailing slash as identical.
The following list contains all such CMP message types. The prefixed
numbers reflect the ASN.1 numbering of the respective element.

```
    [0] Initialization Request
    [2] Certification Request
    [4] PKCS-10 Request
    [6] pop Response
    [7] Key Update Request
    [9] Key Recovery Request
    [11] Revocation Request
    [13] Cross-Certification Request
    [15] CA Key Update Announcement
    [16] Certificate Announcement
    [17] Revocation Announcement
    [18] CRL Announcement
    [20] Nested Message
    [21] General Message
    [23] Error Message
    [24] Certificate Confirmation
    [25] Polling Request
```

An example of a Request-Line and a Host header field in an HTTP/1.1
header, sending a CMP request to a server, located in the "/cmp"
directory of the host example.com, would be

```
    POST /cmp HTTP/1.1
    Host: example.com
```

or in the absoluteURI form

```
    POST http://example.com/cmp/ HTTP/1.1
    Host: example.com
```

A CMP server may be logically located either inside the root- or
within subdirectories of an HTTP server domain.

3.7. Pushing of Announcements

   A CMP server may create event-triggered announcements or generate
   them on a regular basis. It MAY utilize HTTP transport to convey
   them to a suitable recipient. No request message for is specified,
   so they can only be pushed to the recipient. If an EE wants to poll
   for a potential CA Key Update Announcement or the current CRL, a PKI
   Information Request using a General Message as described in E.5 of
   [RFC4210] can be used.

   When pushing Announcements, ASN.1 encoded structures are sent as the
   entity-body of an HTTP POST request.

   Suitable recipients for CMP announcements might e.g. be repositories
   storing the announced information such as directory services. Those
   listen for incoming messages, utilizing the same HTTP Request-URI
   scheme as defined in Section 3.6.

   The following PKIMessages are announcements that may be pushed by a
   CA. The prefixed numbers reflect ASN.1 numbering of the respective
   element.

      [15] CA Key Update Announcement
      [16] Certificate Announcement
      [17] Revocation Announcement
      [18] CRL Announcement

   CMP Announcement messages do not require any CMP response. However,
   the recipient MUST acknowledge receipt with a HTTP response having an
   appropriate status code and an empty body. When not receiving such
   response it MUST be assumed that the delivery was not successful and
   if applicable the sending side may retry sending the Announcement
   after waiting for an appropriate time span.

   If the announced issue was successfully stored in a database or was
   already present, the answer MUST be an HTTP response with a "201
   Created" status code and empty message body.

   In case the announced information was only accepted for further
   processing, the status code of the returned HTTP response MAY also be
   "202 Accepted". After an appropriate delay, the sender may then try
   to send the Announcement again and may repeat this until it receives
   a confirmation that it had been successfully processed. The
   appropriate duration of the delay and the option to increase it
   between consecutive attempts should be carefully considered.

   A receiver MUST answer with a suitable 4xx or 5xx HTTP error code
   when a problem occurs.

3.8. HTTP Considerations

   In general CMP messages are not cachable. HTTP headers of requests
   and responses MUST include "Cache-Control: no-cache" and, in case of
   HTTP/1.0, additionally "Pragma: no-cache" to protect the client from
   getting cached responses.

   Connection management is based on the HTTP provided mechanisms
   ("Connection" and "Proxy-Connection" header fields).

   While implementations MAY make use of all defined features of the
   HTTP protocol, they SHOULD keep the protocol utilization as simple as
   possible. E.g. there is no benefit in using chunked Transfer-
   Encoding as the length of an ASN.1 sequence is know when starting to
   send it.

   There is no need for the clients to send an "Expect" request-header
   field with the "100-continue" expectation and wait for a "100
   Continue" status as described in chapter 8.2.3 of [RFC2616]. The CMP
   payload sent by a client is relatively small, so having extra
   messages exchanged is more inefficient as the server will anyway only
   seldom reject a message without evaluating the body.

   Content codings MAY be applied.

4. **Compatibility Issues with Legacy Implementations**

   As this document was subject of multiple changes during the long
   period of time it was created in, implementations using a different
   approach for HTTP transport may exist. While only those
   implementations according to this specification are compliant,
   implementers should to be aware that there might be existing ones
   which behave differently.

   Legacy implementations might also use an unregistered "application/
   pkixcmp-poll" MIME type as it was specified in earlier drafts of this
   document. Here, the entity-body of an HTTP POST request contains the
   DER-encoded PKIMessage prefixed by an additional so-called TCP-
   Message field. The "TCP-Based Management Protocol" specifying those
   TCP-Messages has been described in draft versions of this document
   but was removed.

5. Security Considerations

   The following aspects need to be considered by server side
   implementers:

   1. There is the risk for denial of service attacks through resource
       consumption by opening many connections. Therefore idle
       connections should be terminated after an appropriate timeout,
       maybe also depending on the available free resources. After
       sending a CMP Error Message, the server should close the
       connection even if the CMP transaction is not yet fully
       completed.

   2. There is no security at the HTTP protocol level (unless tunneled
       via TLS) and thus information from the HTTP protocol SHOULD NOT
       be used to change state of the transaction. Change of state
       SHOULD be triggered by signed PKIMessages only. Likewise, the
       clients SHOULD NOT support the 301 HTTP status code which could
       otherwise be used by an man-in-the-middle attacker to block them
       permanently from contacting the correct server.

   3. If no measures to authenticate and protect the HTTP responses to
       pushed Announcement messages are in place their information
       regarding the Announcement's processing state may not be trusted.
       In that case the overall design of the PKI system must not depend
       on the Announcements being reliably received and processed by
       their destination.

6. Information Security Considerations

   CMP provides inbuilt integrity protection and authentication. Due to
   the nature of a PKI, from a security perspective the information
   communicated unencrypted does not contain sensitive information.

   However, it might be possible for an interceptor to utilize the
   available information to gather confidential technical or business
   critical information. Therefore users of the HTTP CMP transport
   might want to use HTTP over TLS according to [RFC2818] or should
   consider to use virtual private networks created e.g. by utilizing
   Internet Protocol Security according to [RFC4301].

7. IANA Considerations

The IANA has already registered the MIME media type "application/
pkixcmp" for identifying CMP sequences due to an request made in
connection with [RFC2510].

No further action by the IANA is necessary for this document or any
anticipated updates.

[8](#). Acknowledgments

   Until the fifth draft version of this document, released on November
   24th 2000, the sole authors were Amit Kapoor and Ronald Tschlaer from
   Certicom. Up to this point the now removed TCP-Based transport was
   described in detail. They are not available for this working on this
   document anymore at the time it is entering the "Authors Final Review
   state AUTH48". As they therefore cannot approve this document as it
   would be necessary, their names were moved to this section. Their
   contact data as originally stated by them is as follows:

      Amit Kapoor
      Certicom
      25801 Industrial Blvd
      Hayward, CA
      US
      Email: amit@trustpoint.com

      Ronald Tschalaer
      Certicom
      25801 Industrial Blvd
      Hayward, CA
      US
      Email: ronald@trustpoint.com

   The authors gratefully acknowledge the contributions of various
   members of the IETF PKIX Working Group and the ICSA CA-talk mailing
   list (a list solely devoted to discussing CMP interoperability
   efforts).

   By providing ideas, giving hints and doing invaluable review work,
   the following alphabetically listed individuals have significantly
   contributed to this document:

      Tomas Gustavsson, Primekey
      Peter Gutmann, University of Auckland
      Wolf-Dietrich Moeller, Nokia Siemens Networks

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.

[RFC2616]   Fielding, R., Gettys, J., Mogul, J., Frystyk, H.,
            Masinter, L., Leach, P., and T. Berners-Lee, "Hypertext
            Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999.

[RFC3986]   Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform
            Resource Identifier (URI): Generic Syntax", STD 66,
            RFC 3986, January 2005.

[RFC4210]   Adams, C., Farrell, S., Kause, T., and T. Mononen,
            "Internet X.509 Public Key Infrastructure Certificate
            Management Protocol (CMP)", RFC 4210, September 2005.

9.2. Informative References

[RFC1945]   Berners-Lee, T., Fielding, R., and H. Nielsen, "Hypertext
            Transfer Protocol -- HTTP/1.0", RFC 1945, May 1996.

[RFC2510]   Adams, C. and S. Farrell, "Internet X.509 Public Key
            Infrastructure Certificate Management Protocols",
            RFC 2510, March 1999.

[RFC2818]   Rescorla, E., "HTTP Over TLS", RFC 2818, May 2000.

[RFC4301]   Kent, S. and K. Seo, "Security Architecture for the
            Internet Protocol", RFC 4301, December 2005.

Authors' Addresses

    Tomi Kause
    Tectia Corporation
    Fredrikinkatu 42
    Helsinki 00100
    Finland

    Email: toka@tectia.com

    Martin Peylo
    Nokia Siemens Networks
    Linnoitustie 6
    Espoo 02600
    Finland

    Email: martin.peylo@nsn.com