Network Working Group T. Kause
Internet-Draft Tectia
Updates: 4210 (if approved) M. Peylo
Intended status: Standards Track NSN
Expires: December 17, 2011 June 15, 2011

Internet X.509 Public Key Infrastructure -- Transport Protocols for CMP

draft-ietf-pkix-cmp-transport-protocols-12.txt

Abstract

This document describes how to layer the Certificate Management Protocol over various transport protocols. 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 December 17, 2011.

Copyright Notice

Table of Contents

1. Introduction

The Certificate Management Protocol (CMP) [RFC4210] requires well defined transport mechanisms to enable End Entities, RAs and CAs to pass PKIMessage sequences between them. This document defines the transport mechanisms which were 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 TCP-based transport protocol. Its features are 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.

The current version of the CMP specification 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 transport specification 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 specification now reflects that by exclusively describing HTTP transport.

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

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.

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 consisting of more than just a single request/response pair, the statelessness of HTTP is not blocking its usage as transport protocol for CMP messages.

3.1. HTTP Versions

Either HTTP/1.0 as described in [RFC1945] or HTTP/1.1 as in [RFC2616] MAY be used. Naturally, the newer version should be preferred. To support legacy implementations, both server and client MUST be able to interact with counterparts utilizing the other 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.

In contrast to HTTP/1.1, persistent connections are explicitly negotiated in HTTP/1.0. To avoid the problems described in chapter 19.6.2 in [RFC2616], HTTP/1.0 implementations must not send Keep-Alive when talking to proxies.

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. The HTTP responses with empty message body to pushed CMP Announcement messages also utilize the status codes 201 and 202 to identify if the information was properly processed.

Note that a server may return any 1xx, 3xx, 4xx, or 5xx status code

if the HTTP request needs further handling or is otherwise not acceptable.

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 events or generated 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 reply it MUST be assumed that the request was not successfully delivered to its estination.

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 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. As default, the path should end in a "cmp" directory.

3.7. Announcements

A CMP server may create event-triggered announcements or generate them on a regular basis. It MAY also utilize HTTP transport to convey them to a suitable recipient. They can either be pushed to the recipient or polled from the HTTP CMP server.

3.7.1. Pushing of Announcements

The 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. The sending side should assume the delivery unsuccessful without such reply and retry if applicable 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 issue was only stored for further processing, the status code of the returned HTTP response must be "202 Accepted". After an appropriate delay, the server may then try to send the Announcement again and may repeat this until it receives a confirmation that it had been successfully stored. 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.

3.7.2. Polling of Announcements

As an OPTIONAL feature a CA may provide CA Key Update Announcement, Revocation Announcement and CRL Announcement messages for polling using HTTP GET requests. This is not to be confused with the "Polling Request and Response" mechanism defined by CMP.

The server replies with the requested Announcement as the body of a HTTP response having a 200 status code. If no suitable announcement message is available, an HTTP "404 Not Found" error code MUST be returned.

Query components are formed according to [RFC3986]. Their start is indicated by the first question mark in the Request-URI and they are containing "key=value" pairs. Hexadecimal representations of ASN.1 strings used as value MAY contain lower or upper case letters and are neither grouped nor prefixed.

The given examples are for a self-signed certificate with the common name (OID 2.5.4.3) "Example CA", the keyIdentifier in hexadecimal representation BE911E711EDB685BF94D9B176A1BC715CE51D794 and the serial number 008F8B7E383D88327C.

3.7.2.1. CA Key Update Announcement

When updating its key pair, a CA can produce a CA Key Update Announcement Message that can be made available to the relevant end entities. This is described as "Root CA Key Update" in E.4 of

[RFC4210].

A CMP server may provide this message via an HTTP GET request for the CAKeyUpdAnn.PKI file in the respective server's path. The identification of the old key in question is created according to the Authority Key Identifier as defined in chapter 4.2.1.1 of [RFC5280]. The query component then consists of one single "key=value" pair, having the string "AuthorityKeyIdentifier" as key, and the hexadecimal representation of the ASN.1 AuthorityKeyIdentifier sequence as value.

An example of the query component, when requesting a CA Key Update Announcement Message for an old key identified with the AuthorityKeyIdentifier 303C8014BE911E711EDB685BF94D9B176A1BC715CE51D7

94A119A417301531133011060355040313 0A4578616D706C652043418209008F8B7E3 83D88327C

?AuthorityKeyIdentifier=303C8014BE911E711EDB685BF94D9B176A1BC715CE

51D794A119A417301531133011060355040313 0A4578616D706C65204341820900 8F8B7E383D88327C

3.7.2.2. Revocation Announcement

A CMP server MAY permit subjects to poll for a Revocation Announcement using HTTP means. This enables a subject to determine if its certificate is about to be (or has been) revoked.

The Request-URI of the HTTP GET targets the RevAnn.PKI file in the respective server's path. The query component contains two "key=value" pairs identifying the certificate in question:

- an "issuer" key with the hexadecimal representation of the certificate's issuer's GeneralNames ASN.1 sequence as value
- a "serialNumber" key with the hexadecimal representation of the certificate's serial number

An example of the query component, when requesting a Revocation Announcement of a certificate issued by "Example CA" having the decimal serialNumber 6699 would be:

?issuer=3015311330110603550403130A4578616D706C65204341& serialNumber=1A2B

3.7.2.3. CRL Announcement

A CMP server MAY offer the possibility to poll for the latest CRL

Announcement of a specific CA.

The Request-URI targets the CRLAnn.PKI file. The query component consists of one one "key=value" pair containing an "issuer" key with the hexadecimal representation of the CA's GeneralNames' ASN.1 sequence as value.

An example of a Request-URI for the latest CRL Announcement of "Example CA" from a CMP server located in the "/cmp" directory of the host example.com would be

```
http://example.com/cmp/
CRLAnn.PKI?issuer=30153113301106035504031 30A4578616D706C65204341
```

3.8. HTTP Considerations

In general, CMP messages are not cachable; requests and responses MUST include a "Cache-Control: no-cache" (and, if either side uses HTTP/1.0, a "Pragma: no-cache") to prevent the client from getting cached responses.

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

While an implementation MAY make use of all defined features of the HTTP protocol, it SHOULD keep the protocol utilization as simple as possible.

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 seldomly reject the message without looking at the body.

Content codings MAY be applied.

3.9. 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 "TCP-Messaging"

protocol. TCP-Messaging was described in draft versions of this document but was removed.

4. 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.

5. 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].

6. IANA Considerations

The IANA has already registered TCP and UDP port 829 for "PKIX-3 CA/RA" and the MIME media type "application/pkixcmp" for identifying CMP sequences.

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

7. References

7.1. 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.

[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.

[RFC5321] Klensin, J., "Simple Mail Transfer Protocol", RFC 5321, October 2008.

[RFC5646] Phillips, A. and M. Davis, "Tags for Identifying Languages", BCP 47, RFC 5646, September 2009.

7.2. Informative References

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

[RFC2482] Whistler, K. and G. Adams, "Language Tagging in Unicode Plain Text", RFC 2482, January 1999.

[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.

Appendix A. Acknowledgments

Until the fifth draft version of this document, released in 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 individuals, listed alphabetically, have significantly contributed to this document:

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

Appendix B. Registration of the application/pkixcmp Media Type

To: ietf-types@iana.org
Subject: Registration of MIME media type application/pkixcmp

MIME media type name: application

MIME subtype name: pkixcmp

Required parameters: -

Optional parameters: -

Encoding considerations:

Content may contain arbitrary octet values (the ASN.1 DER encoding of a PKIMessage sequence, as defined in the IETF PKIX Working Group specifications). base64 encoding is required for MIME e-mail; no encoding is necessary for HTTP.

Security considerations:

This MIME type may be used to transport Public-Key Infrastructure (PKI) messages between PKI entities. These messages are defined by the IETF PKIX Working Group and are used to establish and maintain an Internet X.509 PKI. There is no requirement for specific security mechanisms to be applied at this level if the PKI messages themselves are protected as defined in the PKIX specifications.

Interoperability considerations: -

Published specification: this document

Applications which use this media type: Applications using certificate management, operational, or ancillary protocols (as defined by the IETF PKIX Working Group) to send PKI messages via E-Mail or HTTP.

Additional information:

- Magic number (s): -
- File extension (s): ".PKI"
- Macintosh File Type Code (s): -

Person and email address to contact for further information:
Martin Peylo, martin.peylo@nsn.com

Intended usage: COMMON

Author/Change controller: Martin Peylo

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