Network Working Group T.
Kause
Internet-Draft
Tectia
Updates: 4210 (if approved) M.
Peylo
Intended status: Standards Track
NSN
Expires: November 8, 2012 May 07,
2012

Internet X.509 Public Key Infrastructure -- HTTP Transport for CMP
draft-ietf-pkix-cmp-transport-protocols-17.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 will expire on November 8, 2012.

Copyright Notice

Table of Contents

14

1. Introduction

The Certificate Management Protocol (CMP) [RFC4210] requires a well defined transport mechanism to enable End Entities (EEs), Registration Authorities (RAs) and Certification 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 its 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.

The usage of HTTP for transporting CMP messages exclusively uses POST method for requests, effectively tunneling CMP over HTTP. While this is generally considered as bad practice and should not be emulated, there are good reasons to do so for transporting CMP. HTTP is used as it is generally easy to implement and able to traverse network borders utilizing ubiquitous proxies. Most importantly, HTTP is already commonly used in existing CMP implementations. Other HTTP request methods such as GET are not used as PKI management operations can only be triggered using CMP's PKI messages which need to be transported within a POST request.

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

Implementations MUST support HTTP/1.0 [RFC1945], and SHOULD support HTTP/1.1 [RFC2616].

3.2. Persistent Connections

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

3.3. General Form

A 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 response in the body of the HTTP response. The HTTP response status code in this case MUST be 200; other "Successful 2xx" codes MUST NOT be used for this purpose. HTTP responses to pushed CMP Announcement messages (i.e., CA Certificate Announcement, Certificate Announcement, Revocation Announcement, and CRL Announcement) utilize the status codes 201 and 202 to identify whether the received information was processed.

While "Redirection 3xx" status codes MAY be supported by implementations, clients should only be enabled to automatically follow them after careful consideration of possible security implications.

All applicable "Client Error 4xx" or "Server Error 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].

A server implementation MUST handle Request-URI paths with or without a trailing slash as identical.

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" path 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
```

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. As no request messages are specified for those announcements 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 Announcement messages, PKIMessage 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

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.

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 implementers and users:

1. There is the risk for denial of service attacks through resource consumption by opening many connections to an HTTP server. 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. Without being encapsulated in effective security protocols such as TLS [RFC5246] there is no integrity protection at the HTTP protocol level. Therefore information from the HTTP protocol should not be used to change state of the transaction.

3. Client users should be aware that storing the target location of a HTTP response with the "301 Moved Permanently" status code could be exploited by a man-in-the-middle attacker to block them permanently from contacting the correct server.

4. 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.

5. CMP provides inbuilt integrity protection and authentication. The information communicated unencrypted in CMP messages does not contain sensitive information endangering the security of the PKI when intercepted. However, it might be possible for an eavesdropper to utilize the available information to gather confidential technical or business critical information. Therefore users of the HTTP transport for CMP might want to consider using HTTP over TLS according to [RFC2818] or virtual private networks created e.g. by utilizing Internet Protocol Security according to [RFC4301].

6. 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.

7. 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

8. References

8.1. Normative References

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

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

[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.

8.2. Informative References

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

[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.

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

Authors' Addresses

Tomi Kause
Tectia Corporation
Kumpulantie 3
Helsinki 00520
Finland

Email: toka@tectia.com

Martin Peylo
Nokia Siemens Networks
Linnoitustie 6
Espoo 02600
Finland

Email: martin.peylo@nsn.com