CMS over COPS
Last Updated: May 31, 2000

Abstract

This memo describes how to use TLS to secure COPS connections over the Internet.

Please send comments on this document to the rap@iphighway.com mailing list.

1. Introduction

COPS [RFC2748] was designed to distribute cleartext policy information from a centralized Policy Decision Point (PDP) to a set of Policy Enforcement Points (PEP) in the Internet. COPS has native security mechanisms to protect the per-hop integrity of the deployed policy. However, COPS has no end-to-end integrity mechanism, and it offers no way to protect the long-term integrity of policy cached at a PEP.
CMS [CMS] was designed to provide a general security encapsulation mechanism, within the context of store-and-forward systems such as e-mail. As such, it is well suited to the end-to-end integrity problem.

This document describes how to use CMS over COPS to provide end-to-end security features. This is abbreviated CMS/COPS.

1.1. Requirements Terminology

Keywords "MUST", "MUST NOT", "REQUIRED", "SHOULD", "SHOULD NOT" and "MAY" that appear in this document are to be interpreted as described in [RFC2119].

2. Certificate Requirements

Certificates used by COPS itself MUST conform to the PKIX profile [PKIX]. CMS/COPS implementations MUST be PKIX clients, and MUST support a secure mechanism to acquire the root certificate authorities for any certificate chains they consume.

If a subjectAltName extension of type dNSName or iPAddress is present in the certificate, that MUST be used as the server identity. Otherwise, the most specific Common Name attribute in the Subject attribute of the certificate MUST be used.

Matching is performed using the matching rules specified by [PKIX]. If more than one identity of a given type is present in the certificate (e.g. more than one dNSName name, a match in any one of the set is considered acceptable.), the client uses the first name to match. Names may contain the wildcard character * which is considered to match any single domain name component or component fragment. E.g. *.a.com matches foo.a.com but not bar.foo.a.com. f*.com matches foo.com but not bar.com.

In some cases the information identifying the COPS will only be an IP address. In this case, the subjectAltName MUST be present in the certificate, and it MUST include an iPAddress format matching the expected name of the PEP.

3. CMS Object Encapsulation within COPS

CMS objects transferred from client to server (PEP to PDP) are encapsulated in a COPS ClientSI object with C-Type=3. CMS objects transferred from server to client (PDP to PEP) are encapsulated in a COPS Decision object with C-Type=6. The data encapsulated must conform to the COPS for Provisioning specification [COPSPR]. The type of CMS object transported in either of these two encapsulations is the DER encoding of a CMS object of type ContentInfo. See [CMS].

4. PIB Object Encapsulation within CMS

Sometimes it is necessary to transfer signed information between client and server, e.g., so that an audit of already distributed...
policy can identify the issuer, or provide long term integrity protection on cached objects, or some similar function. In this type of circumstance, the signature is part of the policy, not the transport protocol used to convey the information between the client and server. COPS uses CMS objects to provide this and other end-to-end security functions. This section defines the encoding of [COPSPR] defined objects (herein referred to as PIB objects) within a CMS object.

4.1. PIB Objects

The Encapsulated-PIBList object is used to encapsulate COPS PIB objects into a CMS of type ContentInfo. This type of a body is identified by

\[
\text{id-cops-encap-pib-list OBJECT IDENTIFIER ::= \{ xxxx x \}}
\]

The ASN.1 structure corresponding to this new content type is

\[
\text{Encapsulated-PIBList ::= SEQUENCE {}
\text{ encapsulatedItem SEQUENCE OF SIZE(0..MAX) OF COPSObject}
\text{}}
\]

\[
\text{COPSObject ::= CHOICE {}
\text{ rule PIBProvisioningRuleId,}
\text{ prefix PIBProvisioningPrefixId,}
\text{ instance PIBProvisioningInstance,}
\text{ pGError PIBGlobalProvisioningError,}
\text{ pCError PIBClassProvisioningError}
\text{}}
\]

\[
\text{PIBSelector ::= INTEGER(0..255)}
\]

\[
\text{PIBProvisioningRuleId ::= SEQUENCE {}
\text{ sNum PIBSelector,}
\text{ sType PIBSelector,}
\text{ policyRuleId OBJECT IDENTIFIER}
\text{}}
\]

\[
\text{PIBProvisioningPrefixId ::= SEQUENCE {}
\text{ sNum PIBSelector,}
\text{ sType PIBSelector,}
\text{ prefixId OBJECT IDENTIFIER}
\text{}}
\]

\[
\text{PIBProvisioningInstance ::= SEQUENCE {}
\text{ sNum PIBSelector,}
\text{ sType PIBSelector,}
\text{ value OCTET STRING -- Or EXPLICIT SEQUENCE Defined by PIB}
\text{}}
\]

\[
\text{PIBGlobalProvisioningError ::= SEQUENCE {}
\text{}}
\]
PIBClassProvisioningError ::= SEQUENCE {
  sNum      PIBSelector,
  sType     PIBSelector,
  eCode     PIBCode,
  eSubCode  PIBCode
}

PIBCode ::= INTEGER(0..65535)

Encapsulated-PIBList objects are DER encoded to conform with CMS.

The meaning of the attributes in these objects is the following:

  encapsulatedItem. This is a sequence of one or more COPS PIB objects.

When used for COPS-PR, a CMS object encodes at most one PIBList object.

A PIBObject is one of the following object types:

  rule. This conveys the identity of a policy instance (PRID),
       encoded as a PIBProvisioningRuleId.

  prefix. This conveys the prefix of an aggregated family (class)
          of policy rules, encoded as a PIBProvisioningPrefixId.

  instance. This encodes the content of a COPS "BER Encoded Policy
            Instance Data" (EPD) object.

  pGError. This conveys global provisioning error information,
           encoded as a PIBGlobalProvisioningError.

  pCError. This conveys PRC class provisioning error information,
           encoded as a PIBClassProvisioningError.

All the encoded PIB Objects consist of the following attributes:

  sNum. This is the COPS-PR S-Number of the encoded data, as it
        appears in a PIB object. This, together with the sType, identifies the COPS message construction being encoded.

  sType. This is the COPS-PR S-Type of the encoded data, as it
         appears in a PIB object. This, together with the sNum, identifies the COPS message construction being encoded.
PIBProvisioningRuleId consists of the following additional attributes:

- policyRule. This is the OBJECT IDENTIFIER of the policy rule being communicated.

PIBProvisioningPrefixId consists of the following attributes:

- prefixRule. This is an OBJECT IDENTIFIER identifying a set of policies being provisioned. All rule instances sharing the same prefix are specified by this object.

PIBProvisioningInstance consists of the following attributes:

- value. This is DER encoded data value of a "BER Encoded Policy Instance Data" (EPD) object [COPS-PR]. That is, the COPS-PR PRC value. The PRC value in a COPS-PR message is BER encoded. This ASN.1 data object is then DER encoded as a SEQUENCE for inclusion in the PIBData object, to accommodate the requirements for CMS.

PIBGlobalProvisioningError consists of the following attributes:

- eCode. This is the primary COPS-PR error code.
- eSubCode. This is the secondary COPS-PR error code (sub-code).

PIBClassProvisioningError consists of the following attributes:

- eCode. This is the primary COPS-PR error code.
- eSubCode. This is the secondary COPS-PR error code (sub-code).

4.2. CMS Cryptographic Operations

The COPS CMS object MAY utilize the SignedData type, EncryptedData type, or both. When using either, the eContent attribute within the encapContentInfo attribute is encoded as defined by Encapsulated-PIBList.

Where PIBData are encapsulated within a COPS CMS object, the eContentType of the EncryptedContentInfo MUST be id-data [CMS].

The signing and encryption algorithms supported MUST be those specified in sections 2.2 and 2.3 of [CMS].

A COPS CMS object MAY contain both public key certificates (Certificate) and attribute certificates (AttributeCertificate). Conforming implementations MUST NOT support other legacy certificate formats. Implementations MUST support the Certificate structure, and SHOULD support AttributeCertificate structure as defined in the PKIX attribute certificate profile [PKIX-ATTRIBUTE].
4.3. CMS Object Verification

An implementation SHOULD verify the signature of a signed COPS CMS object on receipt. In this case, the eContent attribute of the EncapsulatedContentInfo structure MUST NOT be absent. The signature is computed over the entire Encapsulated-PIBList content of the CMS object. If the signature cannot be verified correctly, a COPS Error object with error code set to CMS Authentication Failure (Error-Code 16) MUST be returned.

When an implementation receives a COPS CMS object containing encrypted Encapsulated-PIBList within the CMS EnvelopedData, then the recipient MUST check to see if it is on the list of recipients specified in the RecipientInfos of the EnvelopedData. If not, the recipient MUST indicate an error. If the recipient is in the RecipientInfos and an error occurs during decryption, then the recipient MUST indicate an error.

A COPS CMS object MAY also contain a certs-only CMS structure, or a Certificate Management Message structure, both of which are degenerate forms of CMS structure containing only PKI related information. This use of the CMS object thus can be used to "push" public key and attribute certificates and CRLs using COPS, or to support certificate enrollment. This can be useful in environments where repositories (e.g. LDAP servers) are either not used or not available (e.g. due to crossing a domain boundary). Conforming implementations MUST be able to emit a certs-only CMS structure which contains relevant PKI related information and MUST be able to process a CMS object containing a certs-only CMS structure. The recipient of such a CMS object MUST NOT use the PKI related information without first verifying it, e.g. by checking that issuer’s are trusted, signatures verify, etc.

When the CMS object contains certificates in the certificates attribute of the SignedData, a CRL MAY also be provided in the crls attribute of the SignedData. Implementations MAY use such CRLs to assist in determining whether a certificate has been revoked. Optionally, COPS implementations MAY check the status of certificates using another mechanism, such as Online Certificate Status Protocol [OCSP].

5. Security Considerations

This entire document concerns security.

6. Acknowledgements

This document is loosely based on [STRONG-CRYPTO] by Pat Calhoun, William Bully, and Stephen Farrell. Discussions with David Durham, Keith McCloghrie, and Ylian Sainte-Hillaire also lead to improvements in this document.

7. References


[STRONG-CRYPTO] draft-calhoun-diameter-strong-crypto-xx.txt

8. Author Addresses

Jesse R. Walker
Intel Corporation
2111 N.E. 25th Avenue
Hillsboro, OR 97214
USA
jesse.walker@intel.com

9. Full Copyright Statement

Copyright (C) The Internet Society (2000). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. In addition, the ASN.1 modules presented in Appendices A and B may be used in whole or in part without inclusion of the copyright notice. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process
shall be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns. This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.