INTERNET DRAFT
Intended Status: Informational
Expires: 23 July 2008

R. Housley
Vigil Security

23 January 2008

Digital Signatures on Internet-Draft Documents
<draft-housley-internet-draft-sig-file-00.txt>

Status of this Memo

Abstract

This document specifies the conventions for digital signatures on Internet-Drafts. The Cryptographic Message Syntax (CMS) is used to create a detached signature, which is stored in a separate companion file so that no existing utilities are impacted by the addition of the digital signature.

1. Introduction

This document specifies the conventions for storing a digital signature on Internet-Drafts. The Cryptographic Message Syntax (CMS) [CMS] is used to create a detached signature. The signature is stored in a separate companion file so that no existing utilities are impacted by the addition of the digital signature.

At the time the IETF Secretariat posts the Internet-Draft in the repository, the digital signature is generated and posted as a companion file in the same repository. The digital signature allows anyone to confirm that the contents of the Internet-Draft have not been altered since the time that the document was posted in the repository.

1.1. Terminology

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 [STDWORDS].

1.2. ASN.1

The CMS uses Abstract Syntax Notation One (ASN.1) [X.680]. ASN.1 is a formal notation used for describing data protocols, regardless of the programming language used by the implementation. Encoding rules describe how the values defined in ASN.1 will be represented for transmission. The Basic Encoding Rules (BER) [X.690] are the most widely employed rule set, but they offer more than one way to represent data structures. For example, definite length encoding and indefinite length encoding are supported. This flexibility is not desirable when digital signatures are used. As a result, the Distinguished Encoding Rules (DER) [X.690] were invented. DER is a subset of BER that ensures a single way to represent a given value. For example, DER always employs definite length encoding.

2. Internet-Draft Signature File

All Internet-Draft file names begin with "draft-". The next portion of the file name depends on the source of the document. For example, documents from IETF working groups will have "ietf-" followed by the working group abbreviation, and this is followed by a string that helps people figure out the subject of the document.

All Internet-Draft file names end with a hyphen followed by a two digit version number and a suffix. The suffix indicates the type of file. A plain text file with a suffix of ".txt" is required. Other formats may also be provided, and they employ the appropriate suffix

for the file format.

The companion signature file has exactly the same file name as the Internet-Draft, except that ".sig" is added to the end. Here are a few example names:

Internet-Draft: draft-ietf-example-widgets-03.txt
Signature File: draft-ietf-example-widgets-03.txt.sig

Internet-Draft: draft-ietf-example-widgets-03.ps
Signature File: draft-ietf-example-widgets-03.ps.sig

Internet-Draft: draft-housley-internet-draft-sig-file-00.txt
Signature File: draft-housley-internet-draft-sig-file-00.txt.sig

The IETF Secretariat will post the signature file in the repository at the same time that the Internet-Draft is posted.

3.1. Need for Canonicalization of Text Files

In general, the content of the Internet-Draft is treated like a single octet string for the generation of the digital signature. Unfortunately, the plain text file requires canonicalization to avoid signature validation problems. The primary concern is the manner in which different operating systems indicate the end of a line of text. Some systems use a single new-line character, and other systems use the combination of the carriage-return character followed by a line-feed character. For the digital signature to validate properly, a single convention must be employed.

3.2. Canonicalization

The canonicalization procedure follows the conventions used for text files in the File Transfer Protocol (FTP) [FTP]. Such files must be supported by FTP implementations, so code reuse seems likely.

The canonicalization procedure converts the data from its internal character representation to the standard 8-bit NVT-ASCII representation (see TELNET [TELNET]). In accordance with the NVT standard, the <CRLF> sequence MUST be used to denote the end of a line of text. Using the standard NVT-ASCII representation means that data MUST be interpreted as 8-bit bytes.

Other nonprintable characters, such as tab, form-feed, and backspace, do occur in Internet-Drafts, and these characters MUST NOT be changed in any way.

3. CMS Profile

CMS is used to construct the detached signature of the Internet-Draft. The CMS ContentInfo content type MUST always be present, and it MUST encapsulate the CMS SignedData content type. Since a detached signature is being created, the CMS SignedData content type MUST NOT encapsulate the Internet-Draft. The CMS detached signature is summarized by:

```
ContentInfo {
  contentType          id-signedData, -- (1.2.840.113549.1.7.2)
  content              SignedData
}

SignedData {
  version              CMSVersion, -- Always set to 3
  digestAlgorithms     DigestAlgorithmIdentifiers,
  encapContentInfo     EncapsulatedContentInfo,
  certificates         CertificateSet, -- Secretariat certificate
  crls                 CertificateRevocationLists,
  signerInfos          SET OF SignerInfo -- Only one
}

SignerInfo {
  version              CMSVersion, -- Always set to 3
  sid                  SignerIdentifier,
  digestAlgorithm      DigestAlgorithmIdentifier,
  signedAttrs          SignedAttributes, -- Always present
  signatureAlgorithm   SignatureAlgorithmIdentifier,
```

```
  signature             SignatureValue,
  unsignedAttrs         UnsignedAttributes -- Optional
}

EncapsulatedContentInfo {
  eContentType          id-asciiTextWithCRLF, -- (TBD)
  eContent              OCTET STRING      -- Always absent
}
```

3.1. ContentInfo

The CMS requires the outer-most encapsulation to be ContentInfo [CMS]. The fields of ContentInfo are used as follows:

contentType
: indicates the type of the associated content, and for the detached Internet-Draft signature file, the encapsulated type is always SignedData, so the id-signedData (1.2.840.113549.1.7.2) object identifier MUST be present in

this field.

content
: holds the content, and for the detached Internet-Draft signature file, the content is always a SignedData content.

3.2. SignedData

The SignedData content type [CMS] contains the signature of the Internet-Draft and information to aid in the validation of that signature. The fields of SignedData are used as follows:

version
: is the syntax version number, and for this specification, the version number MUST be set to 3.

digestAlgorithms
: is a collection of one-way hash function identifiers. It MUST contain the identifier used by the IETF Secretariat to generate the digital signature. See the discussion of digestAlgorithm in Section 3.2.1.

encapContentInfo
: is the signed content, including a content type identifier. Since a detached signature is being created, it does not encapsulate the Internet-Draft. The use of the EncapsulatedContentInfo type is discussed further in Section 3.2.2.

certificates
: is an optional collection of certificates. It SHOULD include the X.509 certificate needed to validate the digital signature value. Certification Authority (CA) certificates and end entity certificates MUST conform to the certificate profile specified in [PKIX1].

crls
: is an optional collection of certificate revocation lists (CRLs). It SHOULD NOT include any CRLs; however, any CRLs that are present MUST conform to the CRL profile specified in [PKIX1].

signerInfos
: is a collection of per-signer information, and for this specification, the collection must contain exactly one SignerInfo that represents the IETF Secretariat. The use of the SignerInfo type is discussed further in Section 3.2.1.

3.2.1. SignerInfo

The IETF Secretariat is represented in the SignerInfo type. The fields of SignerInfo are used as follows:

version
: is the syntax version number. In this specification, the version MUST be set to 3.

sid
: identifies the IETF Secretariat's public key. In this specification, the subjectKeyIdentifier alternative is always used, which identifies the public key directly. This identifier MUST match the value included in the subjectKeyIdentifier certificate extension in the IETF

Secretariat's X.509 certificate.

digestAlgorithm
identifies the one-way hash function, and any associated parameters, used by the IETF Secretariat to generate the digital signature.

signedAttrs
is an optional set of attributes that are signed along with the content. The signedAttrs are optional in the CMS, but signedAttrs is required for in this specification. The SET OF Attribute must be encoded with the distinguished encoding rules (DER) [X.690]. Section 2.2.3 of this document lists the signed attributes that must be included in the collection. Other signed attributes may also be included.

signatureAlgorithm
identifies the digital signature algorithm, and any associated parameters, used by IETF Secretariat to generate the digital signature.

signature
is the digital signature value generated by the IETF Secretariat.

unsignedAttrs
is an optional set of attributes that are not signed. Unsigned attributes are usually omitted; however, the unsigned attributes MAY hold a trusted timestamp generated in accordance with [TSP]. Section 2.2.4 of [TSP] provides more information about this unsigned attribute.

3.2.2. EncapsulatedContentInfo

The EncapsulatedContentInfo structure contains a content type identifier. Since a detached signature is being created, it does not encapsulate the Internet-Draft. The fields of EncapsulatedContentInfo are used as follows:

eContentType

is an object identifier that uniquely specifies the content type. It MUST contain id-asciiTextWithCRLF (TBD).

eContent
is optional. When an encapsulated signature is generated, the content to be signed is carried in this field. Since a detached signature is being created, eContent MUST be absent.

3.2.3. Signed Attributes

The IETF Secretariat MUST digitally sign a collection of attributes along with the Internet-Draft. Each attribute in the collection MUST be DER-encoded. The syntax for attributes is defined in [X.501], and the X.500 Directory provides a rich attribute syntax. A very simple subset of this syntax is used extensively in [CMS], where ATTRIBUTE.&Type and ATTRIBUTE.&id are the only parts of the ATTRIBUTE class that are employed.

Each of the attributes used with this CMS profile has a single attribute value. Even though the syntax is defined as a SET OF AttributeValue, there MUST be exactly one instance of AttributeValue present.

The SignedAttributes syntax within signerInfo is defined as a SET OF Attribute. The SignedAttributes MUST include only one instance of any particular attribute.

The IETF Secretariat MUST include the content-type, message-digest, and signing-time attributes. The IETF Secretariat MAY also include the binary-signing-time signed attribute as well as any other attribute that is deemed appropriate. The intent is to allow additional signed attributes to be included if a future need is identified. This does not cause an interoperability concern because unrecognized signed attributes are ignored at verification.

3.2.3.1. Content-Type Attribute

A content-type attribute is required to contain the same object identifier as the content type contained in the EncapsulatedContentInfo. The IETF Secretariat MUST include a

content-type attribute containing id-asciiTextWithCRLF (TBD).

Section 11.1 of [CMS] defines the content-type attribute.

3.2.3.2. Message-Digest Attribute

The IETF Secretariat MUST include a message-digest attribute, having as its value the output of a one-way hash function computed on the Internet-Draft that is being signed. Section 11.2 of [CMS] defines the message-digest attribute.

3.2.3.3. Signing-Time Attribute

The IETF Secretariat MUST include signing-time attribute, specifying the time, based on the local system clock, at which the digital signature was applied to the Internet-Draft. Section 11.3 of [CMS] defines the content-type attribute.

3.2.3.4. Binary-Signing-Time Attribute

The IETF Secretariat MAY include a binary-signing-time attribute, specifying the time at which the digital signature was applied to the Internet-Draft. If present, the time that is represented MUST match the time represented in the signing-time attribute. The binary-signing-time attribute is defined in [BinTime].

3.2.4. Unsigned Attributes

Unsigned attributes are usually omitted. However, an unsigned attribute MAY hold a trusted timestamp generated in accordance with [TSP]. The idea is to time-stamp the IETF Secretariat digital signature to prove that it was created before a given time. If the IETF Secretariat's certificate is revoked the time stamp allows a verifier to know whether the signature was created before or after the revocation date. Appendix A of [TSP] defines the signature time-stamp attribute that can be used to time-stamp a digital signature.

4. Security Considerations

The Secretariat MUST protect its private key. The use of a hardware security module is RECOMMENDED because compromise of the Secretariat's private key permits masquerade.

The generation of a public/private key pair for signature operations relies on random number generation. The use of an inadequate pseudo-random number generator (PRNG) can result in little or no security. An attacker may find it much easier to reproduce the PRNG environment that produced the key pair, searching the resulting small set of possibilities, rather than brute force searching the whole private

key space. The generation of quality random numbers is difficult, but [RANDOM] offers important guidance in this area.

The Secretariat should be aware that cryptographic algorithms become weaker with time. As new cryptoanalysis techniques are developed and computing performance improves, the work factor to break a particular digital signature algorithm or one-way hash function will be reduced. Therefore, it SHOULD be possible to migrate these algorithms. That is, Secretariat SHOULD be prepared for the supported algorithms to change over time.

5. References

5.1. Normative References

[CMS] Housley, R., "Cryptographic Message Syntax (CMS)", RFC 3852, July 2004.

[PKIX1] Housley, R., Polk, W., Ford, W., and D. Solo, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 3280, April 2002.

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

[X.680] ITU-T Recommendation X.680: Information Technology - Abstract Syntax Notation One, 1997.

[X.690] ITU-T Recommendation X.690 Information Technology - ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER), 1997.

5.2. Informative References

[BinTime] Housley, R., "BinaryTime: An Alternate Format for Representing Date and Time in ASN.1", RFC 4049, April 2005.

[FTP] Postel, J. and J. Reynolds, "File Transfer Protocol", STD 9, RFC 959, October 1985.

[RANDOM] Eastlake, D., Schiller, J., and S. Crocker, "Randomness Recommendations for Security", RFC 4086, June 2005.

[TELNET] Postel, J. and J. Reynolds, "Telnet Protocol Specification", STD 8, RFC 854, May 1983.

[TSP] Adams, C., Cain, P., Pinkas, D., and R. Zuccherato, "Internet X.509 Public Key Infrastructure Time-Stamp Protocol (TSP)", RFC 3161, August 2001.

[X.501] ITU-T Recommendation X.501: Information Technology - Open Systems Interconnection - The Directory: Models, 1993.

6. Acknowledgements

The idea for the Internet-Draft signature file came from a discussion with Scott Bradner at IETF 69 in Chicago.

7. IANA Considerations

None.

{{{ RFC Editor: Please remove this section prior to publication. }}}

Appendix: To Do

In a future version of this specification, add a section that shows how an open source tool can be used to implement the specification.

Authors' Addresses

Russell Housley
Vigil Security, LLC
918 Spring Knoll Drive
Herndon, VA 20170
USA

EMail: housley@vigilsec.com

Copyright and IPR Statements

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 are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY, THE IETF TRUST AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM 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.

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has

made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at http://www.ietf.org/ipr.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.