                Using OpenPGP keys for TLS authentication
                    draft-ietf-tls-openpgp-keys-11

Status of this Memo

   By submitting this Internet-Draft, each author represents that any
   applicable patent or other IPR claims of which he or she is aware
   have been or will be disclosed, and any of which he or she becomes
   aware will be disclosed, in accordance with Section 6 of BCP 79.

   Internet-Drafts are working documents of the Internet Engineering
   Task Force (IETF), its areas, and its working groups. Note that
   other groups may also distribute working documents as Internet-
   Drafts.

   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."

   The list of current Internet-Drafts can be accessed at
   http://www.ietf.org/ietf/1id-abstracts.txt.

   The list of Internet-Draft Shadow Directories can be accessed at
   http://www.ietf.org/shadow.html.

   This Internet-Draft will expire on February 1, 2007.

Abstract

   This memo proposes extensions to the TLS protocol to support the
   OpenPGP key format. The extensions discussed here include a
   certificate type negotiation mechanism, and the required
   modifications to the TLS Handshake Protocol.

Table of Contents

[1](#). **Introduction**

   The IETF has two sets of standards for public key certificates, one
   set for use of X.509 certificates [PKIX] and one for OpenPGP
   certificates [OpenPGP]. At the time of writing, the TLS [TLS]
   standards are defined to use only X.509 certificates. This document
   specifies a way to negotiate use of OpenPGP certificates for a TLS
   session, and specifies how to transport OpenPGP certificates via TLS.
   The proposed extensions are backward compatible with the current TLS
   specification, so that existing client and server implementations
   that make use of X.509 certificates are not affected.

2. **Terminology**

   The term ``OpenPGP key'' is used in this document as in the OpenPGP
   specification [OpenPGP]. We use the term ``OpenPGP certificate'' to
   refer to OpenPGP keys that are enabled for authentication.

   This document uses the same notation and terminology used in the TLS
   Protocol specification [TLS].

   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. Changes to the Handshake Message Contents

   This section describes the changes to the TLS handshake message
   contents when OpenPGP certificates are to be used for authentication.

3.1. Client Hello

   In order to indicate the support of multiple certificate types
   clients MUST include an extension of type "cert_type" (see Section 5)
   to the extended client hello message. The hello extension mechanism
   is described in [TLSEXT].

   This extension carries a list of supported certificate types the
   client can use, sorted by client preference. This extension MUST be
   omitted if the client only supports X.509 certificates. The
   "extension_data" field of this extension contains a
   CertificateTypeExtension structure.

```
      enum { client, server } ClientOrServerExtension;

      enum { X.509(0), OpenPGP(1), (255) } CertificateType;

      struct {
         select(ClientOrServerExtension) {
            case client:
               CertificateType certificate_types<1..2^8-1>;
            case server:
               CertificateType certificate_type;
         }
      } CertificateTypeExtension;
```

   No new cipher suites are required to use OpenPGP certificates. All
   existing cipher suites that support a compatible, with the key, key
   exchange method can be used in combination with OpenPGP certificates.

3.2. Server Hello

   If the server receives a client hello that contains the "cert_type"
   extension and chooses a cipher suite that requires a certificate,
   then two outcomes are possible. The server MUST either select a
   certificate type from the certificate_types field in the extended
   client hello or terminate the connection with a fatal alert of type
   "unsupported_certificate".

   The certificate type selected by the server is encoded in a
   CertificateTypeExtension structure, which is included in the extended
   server hello message using an extension of type "cert_type". Servers

that only support X.509 certificates MAY omit including the
"cert_type" extension in the extended server hello.

3.3. Server Certificate

The contents of the certificate message sent from server to client
and vice versa are determined by the negotiated certificate type and
the selected cipher suite's key exchange algorithm.

If the OpenPGP certificate type is negotiated then it is required to
present an OpenPGP certificate in the Certificate message. The
certificate must contain a public key that matches the selected key
exchange algorithm, as shown below.

     Key Exchange Algorithm OpenPGP Certificate Type

     RSA   RSA public key which can be used for
                             encryption.

     DHE_DSS   DSS public key which can be used for
                             authentication.

     DHE_RSA   RSA public key which can be used for
                             authentication.

An OpenPGP certificate appearing in the Certificate message is sent
using the binary OpenPGP format. The certificate MUST contain all
the elements required by Section 10.1 of [OpenPGP].

The option is also available to send an OpenPGP fingerprint, instead
of sending the entire certificate. The process of fingerprint
generation is described in section 11.2 of [OpenPGP]. The peer shall
respond with a "certificate_unobtainable" fatal alert if the
certificate with the given fingerprint cannot be found. The
"certificate_unobtainable" fatal alert is defined in section 4 of
[TLSEXT].

```
    enum {
         cert_fingerprint (0), cert (1), (255)
    } OpenPGPCertDescriptorType;

    opaque OpenPGPCertFingerprint<16..20>;

    opaque OpenPGPCert<0..2^24-1>;

    struct {
         OpenPGPCertDescriptorType descriptorType;
         select (descriptorType) {
              case cert_fingerprint: OpenPGPCertFingerprint;
              case cert: OpenPGPCert;
         }
    } Certificate;
```

3.4. Certificate request

The semantics of this message remain the same as in the TLS
specification. However if this message is sent, and the negotiated
certificate type is OpenPGP, the "certificate_authorities" list MUST
be empty.

3.5. Client certificate

This message is only sent in response to the certificate request
message. The client certificate message is sent using the same
formatting as the server certificate message and it is also required
to present a certificate that matches the negotiated certificate
type. If OpenPGP certificates have been selected and no certificate
is available from the client, then a Certificate structure that
contains an empty OpenPGPCert vector MUST be sent. The server SHOULD
respond with a "handshake_failure" fatal alert if client
authentication is required.

3.6. Other Handshake messages

All the other handshake messages are identical to the TLS
specification.

[4](#). **Security Considerations**

   All security considerations discussed in [TLS], [TLSEXT] as well as
   [OpenPGP] apply to this document. Considerations about the use of
   the web of trust or identity and certificate verification procedure
   are outside the scope of this document. These are considered issues
   to be handled by the application layer protocols.

   The protocol for certificate type negotiation is identical in
   operation to ciphersuite negotiation of the [TLS] specification with
   the addition of default values when the extension is omitted. Since
   those omissions have a unique meaning and the same protection is
   applied to the values as with ciphersuites, it is believed that the
   security properties of this negotiation are the same as with
   ciphersuite negotiation.

   When using OpenPGP fingerprints instead of the full certificates, the
   discussion in Section 6.3 of [TLSEXT] for "Client Certificate URLs"
   applies, especially when external servers are used to retrieve keys.
   However a major difference is that while the "client_certificate_url"
   extension allows to identify certificates without including the
   certificate hashes, this is not possible in the protocol proposed
   here. In this protocol the certificates, when not sent, are always
   identified by their fingerprint, which serves as a cryptographic hash
   of the certificate (see Section 11.2 of [OpenPGP]).

   The information that is available to participating parties and
   eavesdroppers (when confidentiality is not available through a
   previous handshake) is the number and the types of certificates they
   hold, plus the contents of certificates.

5. IANA Considerations

   This document defines a new TLS extension, "cert_type", assigned a
   value of TBD-BY-IANA (the value 7 is suggested) from the TLS
   ExtensionType registry defined in [TLSEXT]. This value is used as
   the extension number for the extensions in both the client hello
   message and the server hello message. The new extension type is used
   for certificate type negotiation.

   The "cert_type" extension contains an 8-bit CertificateType field,
   for which a new registry, named "TLS Certificate Types", is
   established in this document, to be maintained by IANA. The registry
   is segmented in the following way:

   1. Values 0 (X.509) and 1 (OpenPGP) are defined in this document.

   2. Values from 2 through 223 decimal inclusive are assigned via IETF
       Consensus [RFC2434].

   3. Values from 224 decimal through 255 decimal inclusive are
       reserved for Private Use [RFC2434].

6. References

6.1. Normative References

   [TLS]   Dierks, T. and E. Rescorla, "The TLS Protocol Version
               1.1", RFC 4346, April 2006.

   [OpenPGP]   Callas, J., Donnerhacke, L., Finey, H., Shaw, D., and R.
               Thayer, "OpenPGP Message Format",
               draft-ietf-openpgp-rfc2440bis-18 (work in progress),
               May 2006.

   [TLSEXT]   Blake-Wilson, S., Nystrom, M., Hopwood, D., Mikkelsen, J.,
               and T. Wright, "Transport Layer Security (TLS)
               Extensions", RFC 4366, April 2006.

   [RFC2434]   Narten, T. and H. Alvestrand, "Guidelines for Writing an
               IANA Considerations Section in RFCs", RFC 2434,
               October 1998.

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

6.2. Informative References

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

Appendix A. Acknowledgements

   This document was based on earlier work made by Will Price and
   Michael Elkins.

   The author wishes to thank Werner Koch, David Taylor, Timo Schulz,
   Pasi Eronen, Jon Callas, Stephen Kent, Robert Sparks and Hilarie
   Orman for their suggestions on improving this document.

Author's Address

Nikos Mavrogiannopoulos
Independent
Arkadias 8
Halandri, Attiki 15234
Greece

Email: nmav@gnutls.org
URI: http://www.gnutls.org/

Intellectual Property Statement

Disclaimer of Validity

Copyright Statement

Acknowledgment