TLS Working Group
Internet-Draft
Expires: December 7, 2006

N. Mavrogiannopoulos
Independent

June 5, 2006

Using OpenPGP keys for TLS authentication
draft-ietf-tls-openpgp-keys-10

Status of this Memo

Copyright Notice

Abstract

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

1. Introduction

At the time of writing, TLS [TLS] uses the PKIX [PKIX] infrastructure, to provide certificate services. Currently the PKIX protocols are limited to a hierarchical key management and as a result, applications which follow different - non hierarchical - trust models, could not be benefited by TLS.

OpenPGP keys (sometimes called OpenPGP certificates), provide security services for electronic communications. They are widely deployed, especially in electronic mail applications, provide public key authentication services, allow distributed key management and can be used with a non hierarchical trust model called the "web of trust" [WOT].

This document will extend the TLS protocol to support OpenPGP keys using the existing TLS cipher suites. In brief this would be achieved by adding a negotiation of the certificate type in addition to the normal handshake negotiations. Then the required modifications to the handshake messages, in order to hold OpenPGP keys as well, will be described. The normal handshake procedure with X.509 certificates is not altered, to preserve compatibility with existing TLS servers and clients.

This document uses the same notation 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].

2. Changes to the Handshake Message Contents

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

2.1. Client Hello

In order to indicate the support of multiple certificate types clients will include an extension of type "cert_type" (see Section 4) 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 will contain 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 keys. All existing cipher suites that support a compatible with the key, key exchange method can be used in combination with OpenPGP keys.

2.2. Server Hello

Servers that receive an extended client hello containing the "cert_type" extension, and have chosen a cipher suite that supports certificates, they MUST 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.

2.3. 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 key in the Certificate message. The OpenPGP key must contain a public key that matches the selected key exchange algorithm, as shown below.

Key Exchange Algorithm	OpenPGP Key Type
RSA	RSA public key which can be used for encryption.
DHE_DSS	DSS public key.
DHE_RSA	RSA public key which can be used for signing.

An OpenPGP public key appearing in the Certificate message will be sent using the binary OpenPGP format. The term public key is used to describe a composition of OpenPGP packets to form a block of data which contains all information needed by the peer. This includes public key packets, user ID packets and all the fields described in section 10.1 of [OpenPGP].

The option is also available to send an OpenPGP fingerprint, instead of sending the entire key. 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 key with the given key fingerprint cannot be found. The "certificate_unobtainable" fatal alert is defined in section 4 of [TLSEXT].

If the key is not valid, expired, revoked, corrupt, the appropriate fatal alert message is sent from section A.3 of the TLS specification. If a key is valid and neither expired nor revoked, it is accepted by the protocol. The key validation procedure is a local matter outside the scope of this document.

```
enum {
     key_fingerprint (0), key (1), (255)
} PGPKeyDescriptorType;

opaque PGPKeyFingerprint<16..20>;

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

struct {
     PGPKeyDescriptorType descriptorType;
     select (descriptorType) {
          case key_fingerprint: PGPKeyFingerprint;
          case key: PGPKey;
     }
} Certificate;
```

2.4. 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.

2.5. 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 keys have been selected, and no key is available from the client, then a Certificate that contains an empty PGPKey should be sent. The server may respond with a "handshake_failure" fatal alert if client authentication is required.

2.6. Other Handshake messages

The rest of the handshake messages such as the server key exchange, the certificate verify and the finished messages are identical to the TLS specification.

3. Security Considerations

As with X.509 ASN.1 formatted keys, OpenPGP keys need specialized parsers. Care must be taken to make those parsers safe against maliciously modified keys, that could cause arbitrary code execution.

Security considerations about the use of the web of trust or the verification procedure are outside the scope of this document and they are considered an issue to be handled by local policy.

4. 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 will be 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].

5. References

5.1. 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., and R. Thayer, "OpenPGP Message Format", RFC 2440, November 1998.

[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.

5.2. 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.

[WOT] Abdul-Rahman, A., "The PGP Trust Model", EDI Forum: The Journal of Electronic Commerce, April 1997.

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 and Pasi Eronen 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

Funding for the RFC Editor function is currently provided by the Internet Society.