Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: February 5, 2011

P. Hoffman
VPN Consortium

August 4, 2010

Using Secure DNS to Retrieve Keys Used for Authenticating TLS Servers
draft-hoffman-tls-keys-from-dns-00

Abstract

TLS requires the use of PKIX certificates for authenticating the server. Some people want to obtain the public key used in this authentication using other methods. This document describes how to securely retrieve a TLS server's public key from the DNS.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

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."

This Internet-Draft will expire on February 5, 2011.

1. Introduction

The first response from the server in TLS [RFC5246] contains a PKIX certificate. In order for the TLS client to authenticate that it is talking to the expected TLS server, the client must validate that the key in this certificate is associated with the domain name used by the client to get to the server. To do this, the client must extract the domain name from one of many places in the PKIX certificate, must trust the trust anchor upon which the server's PKIX certificate is rooted, and must7 perform correct PKIX validation on the certificate.

Some people want a different way to authenticate the server without using PKIX. In order to do this, the TLS client must have a copy of the TLS server's key that was received in a trusted fashion, and a trusted belief that the key is associated with the domain name used to reach the TLS server. This key and association can be gotten out of band, but a more scalable way to get them is by using the DNS.

DNSSEC, which is defined in RFCs 4033, 4034, and 4035 ([RFC4033], [RFC4034], and [RFC4035]), uses cryptographic keys and digital signatures to provide authentication of DNS data. Information retrieved from the DNS and that is validated using DNSSEC is thereby proved to be the authoritative data.

This document defines a secure method to get a key usable in TLS for a particular domain name using DNS protected by DNSSEC. Because the key was retrieved based on a DNS query, the domain name in the query is by definition associated with the key. This document also defines different ways that the key can be used in 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 RFC 2119 [RFC2119].

NOTE TO READERS: THIS DOCUMENT HAS KNOWN HOLES AND ONE GLARING PROBLEM, NAMELY THAT THERE ARE TWO METHODS DEFINED WHILE ONLY ONE WILL BE CHOSEN LATER.

2. Getting TLS Keys from the DNS

This section describes two equivalent methods for encoding TLS keys: a new RR called "TLSKEY" and a TXT record that can be emitted when the query has "_TLSKEY" as the leftmost label. Only one of these methods should be selected for the final protocol. When that decision is made, the method not used will be removed from this document.

2.1. The TLSKEY Resource Record

The new RR TLSKEY is defined here. A query on a domain name for the TLSKEY type can return one or more records of the type TLSKEY.

The format of the TLSKEY response is binary. In the record, all integers consist of two bytes in network byte order. The record, which MUST be in the order defined here, is:

- An integer specifying how many port numbers are listed. This value MUST be at least 1.
- An unordered set of two-byte integers specifying the TCP/UDP ports for which the key is valid. To indicate that the key is valid for all ports on the host associated with the domain name, a single value of 0 MUST be used.
- An integer specifying the type of key.
- A variable-length set of bytes with the key.

[[This will need a proper RRTYPE definiton. That will be added later if this option is chosen.]]

2.2. Using a TXT Resource Record with a _TLSKEY Label Prefix

A request for a TXT RR whose domain is the label _TLSKEY prepended to a domain name can be used to get the KEY associated with the domain name. A query of this can return one or more records of the type TXT.

The format of the TXT response is ASCII text. The record, which MUST be in the order defined here, is:

- One or more instances of "port=" followed by an TCP/UDP port for which the key is valid (expressed as an integer), followd by ";". To indicate that the key is valid for all ports on the host associated with the domain name, a single "port=0;" MUST be used.
- The type of key, specified as "type=nn;" where "nn" is an integer defined below.
- "key=" followed by the set of bytes with the key; the bytes are encoded as lower-case hexadecimal.

2.3. Types of Keys

The initial list of key types is:

- 0 - RSA
- 1 - ECDSA using the P256 curve
- 2 - ECDSA using the P384 curve
- 3 - GOST

[[References are neede above.]]

3. Use of TLS Keys from the DNS in TLS

In order to use one or more TLS keys obtained from the DNS, an application MUST assure that the keys were obtained using DNS protected by DNSSEC. There may be other methods to securely obtain keys in DNS, but those methods are not covered by this document.

An application that requests TLS keys using the method described in the previous section obtain zero or more keys. If the application receives zero keys, it process TLS in the normal fashion.

The application ignores the PKIX certificate received from the server and instead uses the key obtained from the DNS. That is, the client does no processing on the PKIX certificate in the TLS Certificate message. The application instead uses the key as the authenticator.

If the application receives more than one key from the DNS query, it tries each key for which it understands while authenticating the TLS server.

4. IANA Considerations

[[TBD. Will include the registration for the TLSKEY RR if that is the style chosen, as well as a new registry for key types.]]

5. Security Considerations

[[TBD. This section will need to describe, at least, the "attack" where a DNS administrator goes rogue and changes both the A and TLSKEY records for a domain name.]]

6. Normative References

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

[RFC4033] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "DNS Security Introduction and Requirements", RFC 4033, March 2005.

[RFC4034] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Resource Records for the DNS Security Extensions", RFC 4034, March 2005.

[RFC4035] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Protocol Modifications for the DNS Security Extensions", RFC 4035, March 2005.

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

Author's Address

Paul Hoffman
VPN Consortium

Email: paul.hoffman@vpnc.org