Network Working Group	A. Langley	
Internet-Draft	Google Inc	
Intended status: Standards Track	January 20, 2010	
Expires: July 24, 2010		

TOC

Transport Layer Security (TLS) Next Protocol Negotiation Extension
draft-agl-tls-nextprotoneg-00

Abstract

This document describes a Transport Layer Security (TLS) extension for application layer protocol negotiation. This allows the application layer to negotiate which protocol should be performed over the secure connection in a manner which avoids additional round trips and which is independent of the application layer protocols.

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and 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 July 24, 2010.

described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the BSD License.

Table of Contents

---

1. Introduction

TOC

The use of TLS (Dierks, T. and E. Rescorla, “The Transport Layer Security (TLS) Protocol Version 1.2,” August 2008.) [RFC5246] over TCP (Postel, J., “Transmission Control Protocol,” September 1981.) [RFC0793] unavoidably adds at least a round trip to the time taken to establish a connection. By allowing the application layer to perform negotiation during that round trip one gains several advantages:
Firstly, for various reasons, different application layer protocols are increasingly being carried over TLS using a small set of TCP port numbers, most often port 443. Rather than forcing systems administrators to use different IP addresses for every service, negotiation allows multiple services to exist with the same IP address.
Secondly, newer versions of the same application layer protocol can perform discovery and avoid additional round trips after the TLS handshake.
To illustrate the second point, consider HTTP (Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, “Hypertext Transfer Protocol -- HTTP/1.1,” June 1999.) [RFC2616] pipelining. Although it has seen little adoption because of interference from middleware, there is no reason not to use it over TLS where the transport is known to be pristine.
However, even over TLS, the client cannot make pipelined requests until the first reply indicates that the remote peer supports the feature. If the client could negotiate HTTP/1.1 support before the first request, it could start sending pipelined requests immediately.

2. Requirements Notation

TOC

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

3. Next Protocol Negotiation Extension

TOC

A new extension type (next_protocol_negotiation(TBD)) is defined and MAY be included by the client in its ClientHello message. If, and only if, the server sees this extension in the ClientHello, it MAY choose to include the extension in its ServerHello.

```
enum {
  next_protocol_negotiation(TBD), (65535)
} ExtensionType;
```

A new handshake message type (next_protocol(TBD)) is defined. If, and only if, the server included a next_protocol_negotiation extension in its ServerHello message, the client MUST send a NextProtocol message after its ChangeCipherSpec and before its Finished message.

```
struct {
  opaque selected_protocol<0..255>;
  opaque padding<0..255>;
} NextProtocol;
```

The extension_data field of a next_protocol_negotiation in a ClientHello MUST be empty.
The extension_data field in a ServerHello and the NextProtocol message contain opaque bytes to be used by the application layer to negotiate the application layer protocol. The format of this data is not specified in this draft.
Unlike many other TLS extensions, this extension does not establish properties of the session, only of the connection. When session resumption or session tickets (Salowey, J., Zhou, H., Eronen, P., and H. Tschofenig, "Transport Layer Security (TLS) Session Resumption without Server-Side State," January 2008.) [RFC5077] are used, the previous contents of this extension are irrelevant and only the values in the new handshake messages are considered.
For the same reasons, after a handshake has been performed for a given connection, renegotiations on the same connection MUST NOT include the next_protocol_negotiation extension.

4. Security considerations

TOC

The server's list of supported protocols is still advertised in the clear with this extension. This may be undesirable for certain protocols (such as Tor (Dingledine, R., Matthewson, N., and P. Syverson, "Tor: The Second-Generation Onion Router," August 2004.) [tor]) where one could imagine that hostile networks would terminate any TLS connection with a server that advertised such a capability. In this case, clients may wish to opportunistically select a protocol that wasn't advertised by the server. However, the workings of such a scheme are outside the scope of this document.

5. IANA Considerations

TOC

This document requires IANA to update its registry of TLS extensions to assign an entry, referred herein as next_protocol_negotiation. This document also requires IANA to update its registry of TLS handshake types to assign an entry, referred herein as next_protocol.

6. Acknowledgements

TOC

This document benefited specifically from discussions with Wan-Teh Chang and Nagendra Modadugu.

7. References

TOC

7.1. Normative References

TOC

[RFC2119]	Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels," BCP 14, RFC 2119, March 1997 (TXT, HTML, XML).
[RFC5246]	Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2," RFC 5246, August 2008 (TXT).

7.2. Informative References

TOC

[RFC0793]	Postel, J., “Transmission Control Protocol,” STD 7, RFC 793, September 1981 (TXT).
[RFC2616]	Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, “Hypertext Transfer Protocol -- HTTP/1.1,” RFC 2616, June 1999 (TXT, PS, PDF, HTML, XML).
[RFC5077]	Salowey, J., Zhou, H., Eronen, P., and H. Tschofenig, “Transport Layer Security (TLS) Session Resumption without Server-Side State,” RFC 5077, January 2008 (TXT).
[tor]	Dingledine, R., Matthewson, N., and P. Syverson, “Tor: The Second-Generation Onion Router,” August 2004.

Appendix A. Changes

TOC

To be removed by RFC Editor before publication

Author's Address

TOC

	Adam Langley
	Google Inc
Email:	agl@google.com