Network Working Group
Internet-Draft
Expires: February 19, 2011

A. Langley
Google Inc

August 18, 2010

Transport Layer Security (TLS) Next Protocol Negotiation Extension
draft-agl-tls-nextprotoneg-01

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.

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 February 19, 2011.

Copyright Notice

Table of Contents

1. Introduction

As the Internet has evolved, it has become commonplace for hosts to initiate connections based on untrusted and possibly hostile data. HTTP [RFC2616] clients are currently the most widespread example of this as they will fetch URLs based on the contents of untrusted webpages.

Any time that a connection is initiated based on untrusted data there is the possibility of a cross-protocol attack. If the attacker can control the contents of the connection in any way (for example, the requested URL in an HTTP connection) they may be able to encode a valid message in another protocol. The connecting host believes that it is speaking one protocol but the server understands it to be another. The application of Postel's Law exacerbates the issue as many servers will permit gross violations of the expected protocol in order to achieve maximum compatibility with clients.

The WebSockets [websockets] protocol seeks to allow low-latency, full-duplex communication between browsers and HTTP servers. However, it also permits an unprecedented amount of attacker control over the contents of the connection. In order to prevent cross-protocol attacks, a mechanism to assure that both client and server are speaking the same protocol is required. To this end, Next Protocol Negotiation extends the TLS [RFC5246] handshake to permit both parties to agree on their intended protocol.

2. Requirements Notation

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].

3. Next Protocol Negotiation Extension

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 [RFC5077] are used, the previous contents of this extension are irrelevent 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

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 [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

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

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

7. References

7.1. Normative References

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

[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.

[RFC5077] Salowey, J., Zhou, H., Eronen, P., and H. Tschofenig, "Transport Layer Security (TLS) Session Resumption without Server-Side State", RFC 5077, January 2008.

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

7.2. Informative References

[websockets] Hickson, I., "The Web Socket protocol", Internet Draft (work in progress), October 2009.

[tor] Dingledine, R., Matthewson, N., and P. Syverson, "Tor: The Second-Generation Onion Router", August 2004.

Appendix A. Changes

To be removed by RFC Editor before publication

Author's Address

Adam Langley
Google Inc

Email: agl@google.com