Internet Engineering Task Force (IETF)
Request for Comments : 7301
Catégorie : En cours de normalisation
ISSN: 2070-1721
Traduction Claude Brière de L'Isle

S. Friedl, Cisco Systems, Inc.
A. Popov, Microsoft Corp.
A. Langley, Google Inc.
E. Stephan, Orange
juillet 2015

Extension de négociation de protocole de couche application de la sécurité de la couche transport (TLS)

Résumé
Le présent document décrit une extension à la sécurité de couche Transport (TLS) pour la négociation de protocole de couche application au sein de la prise de contact TLS. Pour les instances dans lesquelles plusieurs protocoles d'application sont pris en charge sur le même accès TCP ou UDP, cette extension permet à la couche application de négocier quel protocole sera utilisé sur la connexion TLS.

Statut de ce mémoire
Ceci est un document de l'Internet en cours de normalisation.

Le présent document a été produit par l'équipe d'ingénierie de l'Internet (IETF). Il représente le consensus de la communauté de l'IETF. Il a subi une révision publique et sa publication a été approuvée par le groupe de pilotage de l'ingénierie de l'Internet (IESG). Plus d'informations sur les normes de l'Internet sont disponibles à la paragraphe 2 de la [RFC5741].

Les informations sur le statut actuel du présent document, tout errata, et comment fournir des réactions sur lui peuvent être obtenues à http://www.rfc-editor.org/info/rfc7301

Table des Matières

1. Introduction

De plus en plus, les protocoles de couche application sont encapsulés dans le protocole TLS [RFC5246]. Cette encapsulation permet aux applications d'utiliser les liaisons existantes de communications sûres déjà présentes sur l'accès 443 à travers virtuellement l'infrastructure IP mondiale entière.

Lorsque plusieurs protocoles d'application sont pris en charge sur un seul numéro d'accès côté serveur, comme l'accès 443, le client et le serveur ont besoin de négocier un protocole d'application à utiliser avec chaque connexion. Il est souhaitable

de réaliser cette négociation sans ajouter d'allers-retours réseau entre le client et le serveur, car chaque aller-retour va dégrader le ressenti de l'utilisateur final. De plus, il serait avantageux de permettre un choix de certificat sur la base du protocole d'application négocié.

Le présent document spécifie une extension à TLS qui permet à la couche application de négocier le choix du protocole au sein de la prise de contact TLS. Ce travail était demandé par le groupe de travail HTTPbis pour traiter la négociation de HTTP/2 ([RFC7540]) sur TLS ; cependant, ALPN facilite la négociation de protocoles de couche application arbitraires.

Avec ALPN, le client envoie la liste des protocoles d'application pris en charge au titre du message TLS ClientHello. Le serveur choisit un protocole et envoie le protocole choisi au titre du message TLS ServerHello. La négociation du protocole d'application peut donc être réalisée au sein de la prise de contact TLS, sans ajouter d'allers-retours sur le réseau, et permet au serveur d'associer un certificat différent à chaque protocole d'application, si il le désire.

2. Langage des exigences

Les mots clés "DOIT", "NE DOIT PAS", "EXIGE", "DEVRA", "NE DEVRA PAS", "DEVRAIT", "NE DEVRAIT PAS", "RECOMMANDE", "PEUT", et "FACULTATIF" en majuscules dans ce document sont à interpréter comme décrit dans le BCP 14, [RFC2119].

3. Négociation de protocole de couche application

3.1 Extension de négociation de protocole de couche application

Un nouveau type d'extension ("application_layer_protocol_negotiation(16)") est défini et PEUT être inclus par le client dans son message "ClientHello".

```
enum {
   application_layer_protocol_negotiation(16), (65535)
} ExtensionType;
```

Le champ "extension_data" de l'extension ("application_layer_protocol_negotiation(16)") DEVRA contenir une valeur de "ProtocolNameList".

```
opaque ProtocolName<1..2^8-1>;

struct {
   ProtocolName protocol_name_list<2..2^16-1>
} ProtocolNameList;
```

"ProtocolNameList" contient la liste des protocoles annoncée par le client, en ordre de préférence décroissant. Les protocoles sont désignés par des chaînes d'octets non vides, opaques enregistrées par l'IANA, comme décrit plus en détails à la Section 6 ("Considérations relatives à l'IANA") de ce document. Des chaînes vides NE DOIVENT PAS être incluses et les chaînes d'octets NE DOIVENT PAS être tronquées.

Les serveurs qui reçoivent un ClientHello contenant l'extension "application_layer_protocol_negotiation" PEUVENT retourner une réponse avec le choix des protocoles convenables au client. Le serveur ignorera tout nom de protocole qu'il ne reconnaît pas. Un nouveau type d'extension ServerHello ("application_layer_protocol_negotiation(16)") PEUT être retourné au client dans le message ServerHello étendu. Le champ "extension_data" de l'extension ("application_layer_protocol_negotiation(16)") est structuré de la même façon que décrit ci-dessus pour "extension_data" du client, excepté que la "ProtocolNameList" DOIT contenir exactement un "ProtocolName".

Donc, une prise de contact complète avec l'extension "application_layer_protocol_negotiation" dans les messages ClientHello et ServerHello a le flux suivant (à la différence du paragraphe 7.3 de la [RFC5246]) :

Client		**Serveur**
ClientHello	-------->	ServerHello
(extension ALPN & liste de protocoles)		(extension ALPN & protocoles choisis)
		Certificat*
		ServerKeyExchange*
		CertificateRequest*
	<--------	ServerHelloDone

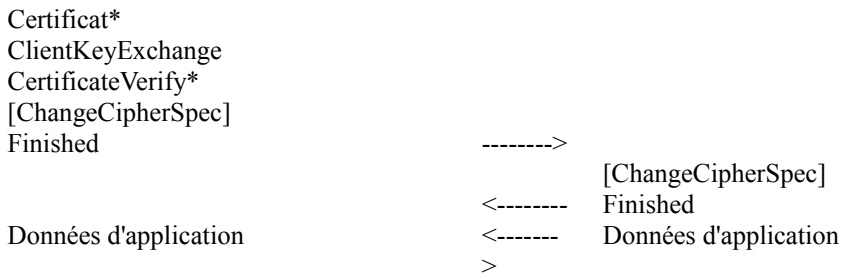

Figure 1

* Indique des messages facultatifs ou dépendants de la situation qui ne sont pas toujours envoyés.

Une prise de contact abrégée avec l'extension "application_layer_protocol_negotiation" a le flux suivant :

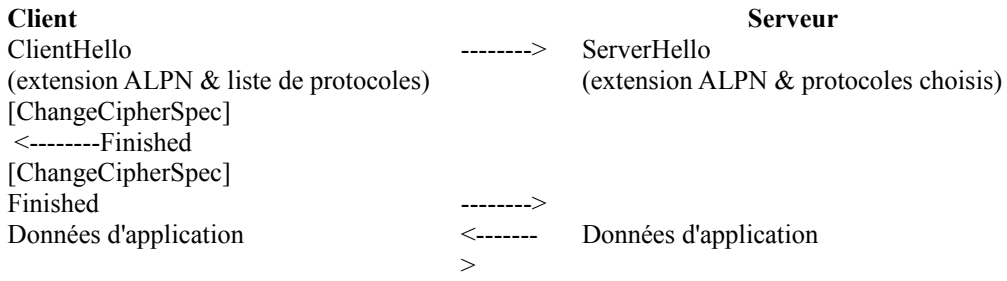

Figure 2

À la différence de beaucoup d'autres extensions à TLS, cette extensions n'établit pas les propriétés de la session, seulement de la connexion. Lorsque on utilise la repris e de session ou des tickets de session [RFC5077], le contenu précédent de cette extension n'est pas pertinent, et seules les valeurs des messages de la nouvelle prise de contact sont pris en compte.

3.2 Choix du protocole

On s'attend à ce qu'un serveur ait une liste de protocoles qu'il prend en charge, dans l'ordre de préférence, et qu'il va seulement choisir un protocole si le client le prend en charge. Dans ce cas, le serveur DEVRAIT choisir le protocole préféré qu'il prend en charge et est aussi annoncé par le client. Au cas où le serveur ne prendrait en charge aucun des protocoles que le client annonce, le serveur DEVRA alors répondre par une alerte fatale "no_application_protocol".

```
enum {
   no_application_protocol(120),
   (255)
} AlertDescription;
```

Le protocole identifié dans le type d'extension "application_layer_protocol_negotiation" dans le ServerHello DEVRA être définitif pour la connexion, jusqu'à ce qu'il soit renégocié. Le serveur NE DEVRA PAS répondre avec un protocole choisi et ensuite utiliser un protocole différent pour l'échange de données d'application.

4. Considérations de conception

L'extension ALPN est destinée à suivre la conception normale d'extension de protocole TLS. Précisément, la négociation est effectuée entièrement au sein de l'échange de hello client/serveur, conformément à l'architecture TLS établie. L'extension de ServerHello "application_layer_protocol_negotiation" est destinée à être définitive pour la connexion (jusqu'à ce que la connexion soit renégociée) et est envoyée en texte source pour permettre aux éléments de réseau de fournir des services différenciés pour la connexion lorsque le numéro d'accès TCP ou UDP n'est pas définitif pour le protocole de couche application à utiliser dans la connexion. En plaçant la responsabilité du choix du protocole chez le serveur, ALPN facilite les scénarios dans lesquels le choix du certificat ou le réacheminement de la connexion peut se fonder sur le protocole négocié.

Finalement, en gérant le choix du protocole en clair au titre de la prise de contact, ALPN évite d'introduire une fausse

confiance à l'égard de la capacité de cacher le protocole négocié avant l'établissement de la connexion. Si il est exigé de cacher le protocole, renégocier après l'établissement de la connexion, ce qui fournira de vraies garanties de sécurité TLS, serait alors une méthodologie préférable.

5. Considérations sur la sécurité

L'extension ALPN n'a pas d'impact sur la sécurité de l'établissement de session TLS ou sur l'échange de données d'application. ALPN sert à fournir un marqueur externe visible pour le protocole de couche application associé à la connexion TLS. Historiquement, le protocole de couche application associé à une connexion pouvait être certifié à partir du numéro d'accès TCP ou UDP utilisé.

Les auteurs de mises en œuvre et éditeurs de document qui ont l'intention d'étendre le registre des identifiants de protocoles en ajoutant de nouveaux identifiants de protocoles devraient considérer que dans les versions TLS 1.2 et en dessous, le client envoie ces identifiants en clair. Ils devraient aussi considérer que, pour au moins les dix années à venir, on s'attend à ce que les navigateurs utiliseront normalement ces versions antérieures de TLS dans le ClientHello initial.

Il faut faire attention lorsque de tels identifiants peuvent laisser échapper des informations personnelles identifiables, ou lorsque de telles fuites peuvent conduire à un profilage ou à la fuite d'informations sensibles. Si cela s'applique à ce nouvel identifiant de protocole, l'identifiant NE DEVRAIT PAS être utilisé dans les configurations TLS où il serait visible en clair, et les documents spécifiant de tels identifiants de protocole DEVRAIENT prévenir contre une telle utilisation non sûre.

6. Considérations relatives à l'IANA

L'IANA a mis à jour son registre "Valeurs de type d'extension" pour inclure les entrées suivantes :

16 application_layer_protocol_negotiation

Le présent document établit un registre d'identifiants de protocole intitulé "Identifiants de protocole de négociation de protocole de couche application (ALPN)" sous l'en-tête existant "Extensions de sécurité de la couche Transport (TLS)".

Les entrées dans ce registre exigent les champs suivants :
- o Protocole : le nom du protocole.
- o Séquence d'identification : ensemble précis de valeurs d'octets qui identifient le protocole. Ce peut être le codage UTF-8 [RFC3629] du nom du protocole.
- o Référence : une référence à la spécification qui définit le protocole.0

Ce registre fonctionne sous la politique de "revue d'expert" définie dans la [RFC5226]. L'expert désigné est invité à encourager l'inclusion d'une référence à une spécification permanente et directement disponible qui permet la création de mises en œuvre interopérables du protocole identifié.

L'ensemble initial d'enregistrements pour ce registre est le suivant :

Protocole : HTTP/1.1
Séquence d'identification : 0x68 0x74 0x74 0x70 0x2f 0x31 0x2e 0x31 ("http/1.1")
Référence : [RFC7230]

Protocole : SPDY/1
Séquence d'identification : 0x73 0x70 0x64 0x79 0x2f 0x31 ("spdy/1")
Référence: http://dev.chromium.org/spdy/spdy-protocol/spdy-protocol-draft1

Protocole : SPDY/2
Séquence d'identification : 0x73 0x70 0x64 0x79 0x2f 0x32 ("spdy/2")
Référence: http://dev.chromium.org/spdy/spdy-protocol/spdy-protocol-draft2

Protocole : SPDY/3
Séquence d'identification : 0x73 0x70 0x64 0x79 0x2f 0x33 ("spdy/3")
Référence : http://dev.chromium.org/spdy/spdy-protocol/spdy-protocol-draft3

7. Remerciements

Le présent document a bénéficié spécifiquement du document d'extension de la négociation du prochain protocole (NPN, *Next Protocol Negotiation*) rédigé par Adam Langley et de discussions avec Tom Wesselman et Cullen Jennings, tous deux de Cisco.

8. Références

8.1 Références normatives

[RFC2119] S. Bradner, "Mots clés à utiliser dans les RFC pour indiquer les niveaux d'exigence", BCP 14, DOI 10.17487/RFC2119, mars 1997. *(MàJ par* RFC8174)

[RFC3629] F. Yergeau, "UTF-8, un format de transformation de la norme ISO 10646", STD 63, novembre 2003.

[RFC5226] T. Narten et H. Alvestrand, "Lignes directrices pour la rédaction d'une section Considérations relatives à l'IANA dans les RFC", BCP 26, mai 2008, DOI 10.17487/RFC5226. (*Remplace* RFC2434 *; remplacée par* RFC8126)

[RFC5246] T. Dierks, E. Rescorla, "Version 1.2 du protocole de sécurité de la couche Transport (TLS)", août 2008. *(P.S.* ; *remplace* RFC3268, 4346, 4366 ; *MàJ* RFC4492 *; rendue obsolète par la* RFC8446)

[RFC7230] R. Fielding, et J. Reschke, "Protocole de transfert Hypertexte (HTTP/1.1) : syntaxe et acheminement de message", juin 2014, DOI 10.17487/RFC7230. *(Remplace* RFC 2145, 2616*, MàJ* RFC 2617, 2618) *(P.S.)*

8.2 Références pour information

[RFC5077] J. Salowey et autres, "Reprise de session de sécurité de la couche Transport (TLS) sans état côté serveur", janvier 2008. (*Remplace* RFC4507) *(P.S. ; rendue obsolète par la* RFC8446)

[RFC7540] M. Belshe, R. Peon, M. Thomson, "HTTPv2", mai 2015. *(P.S.)*

Adresse des auteurs

Stephan Friedl
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134
USA
téléphone : (720)562-6785
mél : sfriedl@cisco.com

Andrei Popov
Microsoft Corp.
One Microsoft Way
Redmond, WA 98052
USA
mél : andreipo@microsoft.com

Adam Langley
Google Inc.
USA
mél : agl@google.com

Emile Stephan
Orange
2 avenue Pierre Marzin
Lannion F-22307
France
mél : emile.stephan@orange.com