Internet Engineering Task Force (IETF)
Request for Comments : 7507
RFC mises à jour : 2246, 4346, 4347, 5246, 6347
Catégorie : En cours de normalisation
ISSN: 2070-1721

B. Moeller
A. Langley
Google

avril 2015
Traduction Claude Brière de L'Isle

Valeur de suite de chiffrement de signalisation (SCSV) de repli pour TLS pour empêcher les attaques de dégradation de protocole

Résumé
Le présent document définit une valeur de signalisation de suite de chiffrement (SCSV, *Signaling Cipher Suite Value*) qui empêche les attaques en dégradation de protocole sur les protocole de sécurité de la couche transport (TLS, *Transport Layer Security*) et sécurité de la couche transport de datagrammes (DTLS, *Datagram Transport Layer Security*). Il met à jour les RFC 2246, 4346, 4347, 5246, et 6347. Les considérations de mise à jour des serveurs sont incluses.

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). Tous les documents approuvés par l'IESG ne sont pas candidats à devenir une norme de l'Internet ; voir la Section 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/rfc7507

Table des Matières

1. Introduction

Pour contourner les problèmes d'interopérabilité avec les serveurs traditionnels, de nombreuses mises en œuvre de client TLS ne s'appuient pas sur le seul mécanisme de négociation de version du protocole TLS mais se reconnectent intentionnellement en utilisant un protocole dégradé si les tentatives initiales de prise de contact échouent. De tels clients peuvent se replier sur des connexions dans lesquelles ils annoncent une version comme TLS 1.0 (ou même son prédécesseur, la couche de connexion sécurisée (SSL, *Secure Socket Layer*) 3.0) comme plus haute version prise en charge.

Bien que de tels essais de repli puissent être utiles en dernier recours pour des connexions avec de vrais serveurs antiques, il

y a un risque que des attaquants actifs puissent exploiter cette stratégie de dégradation pour affaiblir la sécurité du chiffrement des connexions. Aussi, des erreurs de prise de contact dues à des à-coups du réseaux pourraient de la même manière être interprétées à tort comme une interaction avec un serveur traditionnel et résulter en une rétrogradation de protocole.

Toute dégradation de protocole qui n'est pas nécessaire est indésirable (par exemple, de TLS 1.2 à TLS 1.1, si le client et le serveur prennent en fait en charge TLS 1.2) ; la dégradation peut être particulièrement dommageable lorsque le résultat en est la perte des caractéristiques d'extensions de TLS par la rétrogradation à SSL 3.0. Le présent document définit une SCSV qui peut être employée pour empêcher les dégradations involontaires de protocole entre les clients et serveurs qui se conforment au présent document en faisant que le client indique que la tentative de connexion en cours est simplement un repli et que le serveur retourne une alerte fatale si il détecte un repli inapproprié. (L'alerte n'indique pas nécessairement une attaque de dégradation intentionnelle, car des à-coups du réseau pourraient aussi résulter en essais de repli inappropriés.)

La SCSV de repli définie dans le présent document n'est pas un substitut convenable à la négociation appropriée de version de TLS. Les mises en œuvre de TLS doivent traiter correctement la négociation de version TLS et des mécanismes d'extensibilité pour éviter les problèmes de sécurité et les délais de connexion associés aux réessais de repli.

La présente spécification s'applique aux mises en œuvre de TLS 1.0 [RFC2246], TLS 1.1 [RFC4346], et TLS 1.2 [RFC5246], et aux mises en œuvre de DTLS 1.0 [RFC4347] et DTLS 1.2 [RFC6347]. (Elle est particulièrement pertinente si les mises en œuvre de TLS incluent aussi la prise en charge du protocole SSL 3.0 [RFC6101] précédent.) Elle peut de même être appliquée aux versions de protocole ultérieures.

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].

2. Valeurs du protocole

Le présent document définit une nouvelle valeur de suite de chiffrement TLS :

TLS_FALLBACK_SCSV {0x56, 0x00}

C'est une SCSV, c'est-à-dire, elle ne correspond pas réellement à une suite de cryptosystèmes, et elle ne peut jamais être choisie par le serveur dans la prise de contact ; sa présence dans le message Hello du client sert plutôt de signal rétro compatible du client au serveur.

Le présent document alloue aussi une nouvelle valeur d'alerte dans le registre des alertes TLS [RFC5246] :

```
enum {
  /* ... */
  inappropriate_fallback(86),
  /* ... */
  (255)
} AlertDescription;
```

Cette alerte n'est générée que par les serveurs, comme décrit à la Section 3. Elle est toujours fatale.

3. Comportement du serveur

Cette section spécifie le comportement du serveur lors de la réception de la suite de chiffrement TLS_FALLBACK_SCSV provenant d'un client dans une ClientHello.cipher_suites.

- Si TLS_FALLBACK_SCSV apparaît dans ClientHello.cipher_suites et si la plus forte version de protocole prise en charge par le serveur est supérieure à la version indiquée dans le ClientHello.client_version, le serveur DOIT répondre par une alerte fatale "inappropriate_fallback" (sauf si il répond par une alerte fatale "protocol_version" parce que la version indiquée dans le ClientHello.client_version n'est pas prise en charge). Le numéro de version de couche enregistrement pour cette alerte DOIT être réglé à ClientHello.client_version (comme ce serait le cas pour le message Hello du serveur si le serveur continuait la prise de contact) ou au numéro de version de couche enregistrement utilisé par le client.

- Autrement (soit TLS_FALLBACK_SCSV n'apparaît pas, soit il apparaît et la version de protocole du client est au moins la plus forte version de protocole prise en charge par le serveur) le serveur poursuit la prise de contact comme d'habitude.

(Une version de protocole est prise en charge par le serveur si, en réponse aux messages Hello appropriés du client, le serveur va les utiliser pour le ServerHello.server_version. Si une version particulière de protocole est mise en œuvre mais complètement désactivée par les réglages du serveur, elle n'est pas considérée comme prise en charge. Par exemple, si la plus forte version de protocole mise en œuvre est TLS 1.2 mais si l'opérateur du serveur a désactivé cette version, un Hello TLS 1.1 du client avec TLS_FALLBACK_SCSV ne garantit pas une réponse avec une alerte inappropriate_fallback.)

4. Comportement du client

La valeur de suite de chiffrement TLS_FALLBACK_SCSV est destinée à être utilisée par les clients qui répètent une tentative de connexion avec un protocole dégradé (ils effectuent un "réessai de repli") afin de contourner les problèmes d'interopérabilité avec des serveurs traditionnels.

- Si un client envoie un ClientHello.client_version contenant une valeur inférieure à la dernière version (de valeur supérieure) prise en charge par le client, il DEVRAIT inclure la valeur de suite de chiffrement TLS_FALLBACK_SCSV dans le ClientHello.cipher_suites ; voir à la Section 6 les considérations sur la sécurité pour cette recommandation. (Le client DEVRAIT mettre TLS_FALLBACK_SCSV après toutes les suites de chiffrement qu'il a réellement l'intention de négocier.)

- Par exception à ceci, lorsque un client a l'intention de reprendre une session et règle le ClientHello.client_version à la version de protocole négociée pour cette session, il NE DOIT PAS inclure TLS_FALLBACK_SCSV dans ClientHello.cipher_suites. (Dans ce cas, on suppose que le client connaît déjà la plus forte version de protocole acceptée par le serveur : voir l'Appendice E.1 de la [RFC5246].)

- Si un client règle son ClientHello.client_version à sa plus forte version de protocole prise en charge, il NE DOIT PAS inclure TLS_FALLBACK_SCSV dans les ClientHello.cipher_suites.

(Une version de protocole est prise en charge par le client si il tente normalement de l'utiliser dans ses prises de contact. Si une version de protocole particulière est mise en œuvre mais complètement désactivée par les réglages du client, elle n'est pas considérée comme prise en charge. Par exemple, si la plus forte version de protocole de la mise en œuvre est TLS 1.2 mais si l'utilisateur a désactivé cette version, une prise de contact TLS 1.1 sera supposée et ne garantit pas l'envoi du TLS_FALLBACK_SCSV.)

Les essais de repli pourraient être causés par des événements comme des à-coups du réseau, et un client qui inclut le TLS_FALLBACK_SCSV dans les ClientHello.cipher_suites peut recevoir une alerte inappropriate_fallback en réponse, indiquant que le serveur prend en charge une version de protocole supérieure. Donc, si un client a l'intention d'utiliser des réessais pour contourner les à-coups du réseau, il devrait alors réessayer avec la plus forte version qu'il prend en charge.

Si un client garde trace de la plus forte version de protocole apparemment prise en charge par un certain serveur pour l'utiliser ultérieurement dans un ClientHello.client_version, si le client reçoit alors une alerte inappropriate_fallback de ce serveur, il DOIT supprimer la version de protocole prise en charge qu'il a mémorisée. (Sans l'alerte, c'est une bonne idée – mais qui sort du domaine d'application du présent document – pour les clients de supprimer cet état après une temporisation car la plus forte version de protocole d'un serveur peut changer dans le temps.)

Pour les clients qui utilisent le faux départ TLS côté client [RFC7918], il est important de noter que le mécanisme TLS_FALLBACK_SCSV ne peut pas protéger le premier tour de données d'application envoyées par le client : se référer aux Considérations sur la sécurité (Section 6) de la [RFC7918].

5. Considérations de fonctionnement

La mise à jour des grappes de serveurs traditionnels pour ajouter la prise en charge simultanée de nouvelles versions de protocole et de TLS_FALLBACK_SCSV peut subir des complications si la mise en œuvre de serveur traditionnel n'est pas "tolérante aux nouvelles version" (ne peut pas traiter correctement les messages Hello de client pour les nouvelles versions de protocole) : les réessais de repli exigés pour l'interopérabilité avec de plus vieux nœuds serveurs pourraient être rejetés par les nœuds serveurs mis à jour.

La mise à jour de la grappe serveur en deux étapes consécutives rend cela sûr : d'abord, mettre à jour le logiciel du serveur

mais laisser inchangée la plus forte version prise en charge (en désactivant les nouvelles versions dans les réglages du serveur) ; ensuite, après que toutes les mises en œuvre traditionnelles (intolérantes aux nouvelles versions) ont été retirées, changer les réglages de serveur pour permettre les nouvelles versions de protocole.

6. Considérations sur la sécurité

La Section 4 n'exige pas qu'une mise en œuvre de client envoie une TLS_FALLBACK_SCSV dans des cas particuliers, elle le recommande simplement ; le comportement peut être adapté selon les besoins de sécurité du client. Il est important de se rappeler qu'omettre TLS_FALLBACK_SCSV permet les attaques en dégradation, de sorte que les mises en œuvre doivent prendre en compte si la version de protocole donnée par le ClientHello.client_version fournit un niveau de protection acceptable. Par exemple, durant le déploiement initial d'une nouvelle version de protocole (moment où on peut s'attendre à des problèmes d'interopérabilité) un repli en douceur à la précédente version de protocole en cas de problèmes peut être préférable au risque de ne pas être capable de se connecter du tout : de sorte que TLS_FALLBACK_SCSV pourrait être omis pour cette étape de dégradation de protocole particulière.

Cependant, il est fortement recommandé d'envoyer TLS_FALLBACK_SCSV lors d'un repli sur SSL 3.0 car les suites de chiffrement en mode de chaînage de bloc de chiffrement (CBC, *Cipher Block Chaining*) dans SSL 3.0 ont des faiblesses qui ne peuvent pas être traitées par la mise en œuvre de contournements comme les faiblesses restantes dans les versions de protocole (TLS) ultérieures.

7. Considérations relatives à l'IANA

L'IANA a ajouté le numéro de suite de chiffrement TLS 0x56,0x00 du nom de TLS_FALLBACK_SCSV au registre des suites de chiffrement TLS et le numéro d'alerte 86 du nom de inappropriate_fallback au registre des alertes TLS, comme montré ci-dessous. Les registres sont disponibles à <http://www.iana.org/assignments/tls-parameters>.

Ajout au registre TLS Cipher Suite :

Valeur	Description	DTLS-OK	Référence
0x56,0x00	TLS_FALLBACK_SCSV	Oui	RFC 7507

Ajout au registre TLS Alert :

Valeur	Description	DTLS-OK	Référence
86	inappropriate_fallback	Oui	RFC 7507

8. Références

8.1 Références normatives

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

[RFC2246] T. Dierks et C. Allen, "Protocole TLS version 1.0", janvier 1999. *(P.S. ; MàJ par* RFC7919)

[RFC4346] T. Dierks et E. Rescorla, "Protocole de sécurité de la couche Transport (TLS) version 1.1", avril 2006. *(Remplace* RFC2246 *; Remplacée par* RFC5246 *; MàJ par* RFC4366, 4680, 4681, 5746, 6176, 7465, 7507, 7919)

[RFC4347] E. Rescorla, N. Modadugu, "Sécurité de la couche de transport de datagrammes ", avril 2006. *(P.S.)*

[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)

[RFC6347] E. Rescorla, N. Modadugu, "Sécurité de la couche transport de datagrammes, version 1.2", janvier 2012. *(Remplace la* RFC4347) *(P.S. ; MàJ par* RFC7905)

8.2 Références pour information

[RFC6101] A. Freier, P. Karlton, P. Kocher "Protocole de couche de connexion sécurisée (SSL) version 3.0", août 2011. *(Historique)*

[RFC7918] A. Langley, et autres, "Faux départ de sécurité de la couche transport (TLS)", août 2016. *(Information)*

Remerciements

La présente spécification a été inspirée par une proposition antérieure de Eric Rescorla. Nous remercions aussi Daniel Kahn Gillmor, Joe Saloway, Brian Smith, Martin Thomson, et d'autres du groupe de travail TLS pour leurs retours et suggestions.

Adresse des auteurs

Bodo Moeller
Google Switzerland GmbH
Brandschenkestrasse 110
Zurich 8002
Switzerland
mél : bmoeller@acm.org

Adam Langley
Google Inc.
345 Spear St
San Francisco, CA 94105
United States
mél : agl@google.com