Groupe de travail Réseau
Request for Comments : 2818
Catégorie : Information

E. Rescorla, RTFM, Inc.
mai 2000
Traduction Claude Brière de L'Isle

HTTP sur TLS

Statut du présent mémoire
Le présent mémoire apporte des information pour la communauté de l'Internet. Il ne spécifie aucune forme de norme de l'Internet. La distribution du présent mémoire n'est soumise à aucune restriction.

Résumé
Le présent mémoire décrit comment utiliser TLS pour sécuriser les connexions HTTP sur l'Internet. La pratique actuelle est de mettre HTTP en couche sur SSL (le prédécesseur de TLS) en distinguant le trafic sécurisé de celui qui ne l'est pas par l'utilisation d'un accès serveur différent. Le présent document précise cette pratique en utilisant TLS. Un document voisin décrit une méthode pour utiliser HTTP/TLS sur le même accès que l'HTTP normal [RFC2817].

Table des Matières

1. Introduction

HTTP [RFC2616] était à l'origine utilisé en clair sur l'Internet. Cependant, l'utilisation accrue de HTTP pour des applications sensibles a exigé des mesures de sécurité. SSL, et son successeur TLS [RFC2246] ont été conçus pour fournir une sécurité fondée sur le canal. Le présent document décrit comment utiliser HTTP sur TLS.

1.1 Terminologie des exigences

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

2. HTTP sur TLS

Conceptuellement, HTTP/TLS est très simple. On utilise simplement HTTP sur TLS précisément comme on utiliserait HTTP sur TCP.

2.1 Initialisation de connexion

L'agent qui agit comme client HTTP devrait aussi agir comme client TLS. Il devrait initier une connexion au serveur sur l'accès approprié et envoyer ensuite le ClientHello TLS pour commencer la prise de contact *(handshake)* TLS. Lorsque la prise de contact TLS est terminée, le client peut alors initier la première demande HTTP. Toutes les données HTTP DOIVENT être envoyées comme "données d'application" TLS. Le comportement normal HTTP, incluant les connexions retenues, devrait être suivi.

2.2 Clôture de connexion

TLS fournit une facilité de clôture sûre de connexion. Lorsque une alerte de clôture valide est reçue, une mise en œuvre peut être assurée qu'aucune autre donnée ne sera reçue sur cette connexion. Les mises en œuvre de TLS DOIVENT initier un échange d'alertes de clôture avant de clore une connexion. Une mise en œuvre de TLS PEUT, après l'envoi d'une alerte de clôture, fermer la connexion sans attendre que l'homologue envoie son alerte de clôture, générant une "clôture incomplète". Noter qu'une mise en œuvre qui fait cela PEUT choisir de réutiliser la session. Ceci NE DEVRAIT être fait que lorsque l'application sait (normalement par la détection des limites de message HTTP) qu'elle a reçu toutes les données de message dont elle se soucie.

Comme spécifié dans la [RFC2246], toute mise en œuvre qui reçoit une clôture de connexion sans avoir d'abord reçu une alerte de clôture valide (une "clôture prématurée") NE DOIT PAS réutiliser cette session. Noter qu'une clôture prématurée ne remet pas en question la sécurité des données déjà reçues, mais indique simplement que les données qui suivent pourraient avoir été tronquées. Parce que TLS ne tient pas compte des limites de demande/réponse HTTP, il est nécessaire d'examiner les données HTTP elles-mêmes (précisément l'en-tête Content-Length) pour déterminer si la troncature s'est produite à l'intérieur d'un message ou entre les messages.

2.2.1 Comportement du client

Parce que HTTP utilise la clôture de connexion pour signaler la fin des données du serveur, les mises en œuvre de client DOIVENT traiter toutes les clôtures prématurées comme des erreurs et les données reçues comme potentiellement tronquées. Alors que dans certains cas le protocole HTTP permet au client de découvrir si une troncature a eu lieu afin que, si il reçoit la réponse complète, il puisse tolérer de telles erreurs suivant le principe "[être] strict à l'envoi et tolérant à la réception" [RFC1958], souvent la troncature n'apparaît pas dans les données du protocole HTTP ; deux cas méritent en particulier une attention spéciale :

- Une réponse HTTP sans en-tête Content-Length. Comme dans cette situation la longueur des données est signalée par une clôture de connexion, une clôture prématurée générée par le serveur ne peut pas être distinguée d'une clôture parasite générée par un attaquant.

- Une réponse HTTP avec un en-tête Content-Length valide close avant que toutes les données aient été lues. Parce que TLS ne fournit pas de protection au niveau du document, il est impossible de déterminer si le serveur a mal calculé la longueur du contenu ou si un attaquant a tronqué la connexion.

Il y a une exception à la règle ci-dessus. Lorsque il rencontre une clôture prématurée, un client DEVRAIT traiter comme achevées toutes les demandes pour lesquelles il a reçu autant de données que spécifié dans l'en-tête Content-Length.

Un client qui détecte une clôture incomplète DEVRAIT récupérer en douceur. Il PEUT reprendre une session TLS close de cette façon.

Les clients DOIVENT envoyer une alerte de clôture avant de clore la connexion. Les clients qui ne sont pas prêts à recevoir plus de données PEUVENT choisir de ne pas attendre l'alerte de clôture du serveur et de clore simplement la connexion, générant ainsi une clôture incomplète du côté serveur.

2.2.2 Comportement du serveur

La RFC 2616 permet à un client HTTP de clore la connexion à tout moment, et exige des serveurs qu'ils récupèrent en douceur. En particulier, les serveurs DEVRAIENT être prêts à recevoir du client une clôture incomplète, car celui-ci peut souvent déterminer quand est la fin des données du serveur. Les serveurs DEVRAIENT vouloir reprendre les sessions TLS closes de cette façon.

Note de mise en œuvre : dans les mises en œuvre HTTP qui n'utilisent pas de connexions persistantes, le serveur s'attend ordinairement à être capable de signaler la fin des données en clôturant la connexion. Lorsque Content-Length est utilisé, le client peut cependant avoir déjà envoyé l'alerte de clôture et abandonné la connexion.

Les serveurs DOIVENT tenter d'initier un échange d'alertes de clôture avec le client avant le clore la connexion. Les serveurs PEUVENT clore la connexion après l'envoi de l'alerte de clôture, générant ainsi une clôture incomplète du côté client.

2.3 Numéro d'accès

Les premières données qu'un serveur HTTP s'attend à recevoir du client sont la production Request-Line. Les premières données qu'un serveur TLS (et donc un serveur HTTP/TLS) s'attend à recevoir sont le ClientHello. Par conséquent, la pratique courante est de faire fonctionner HTTP/TLS sur un accès séparé afin de distinguer quel protocole est utilisé. Lorsque HTTP/TLS fonctionne sur une connexion TCP/IP, l'accès par défaut est 443. Cela n'empêche pas HTTP/TLS de fonctionner sur un autre transport. TLS suppose seulement un flux de données sur une connexion fiable.

2.4 Format d'URI

HTTP/TLS est différencié des URI HTTP en utilisant l'identifiant de protocole "https" à la place de l'identifiant de protocole "http". Un exemple d'URI qui spécifie HTTP/TLS est :

https://www.example.com/~smith/home.html

3. Identification de point d'extrémité

3.1 Identité du serveur

En général, les demandes HTTP/TLS sont générées en déréférençant un URI. Par conséquent, le nom d'hôte du serveur est connu du client. Si le nom d'hôte est disponible, le client DOIT le vérifier par rapport à l'identité du serveur telle que présentée dans le message de certificat du serveur, afin d'empêcher les attaques par interposition.

Si le client a des informations externes sur l'identité attendue du serveur, la vérification du nom d'hôte PEUT être omise. (Par exemple, un client peut se connecter à une machine dont l'adresse et le nom d'hôte sont dynamiques mais le client connaît le certificat que le serveur va présenter.) Dans de tels cas, il est important de rétrécir la portée des certificats acceptables autant que faire se peut afin d'empêcher les attaques par interposition. Dans des cas particuliers, il peut être approprié que le client ignore simplement l'identité du serveur, mais on doit comprendre que cela laisse la connexion sans défense contre une attaque active.

Si une extension subjectAltName de type dNSName est présente, elle DOIT être utilisée comme l'identité. Autrement, le champ Nom commun (le plus spécifique) dans le champ Subject du certificat DOIT être utilisé. Bien que l'utilisation du nom commun soit une pratique existante, elle est déconseillée et les autorités de certification sont encouragées à utiliser à la place le dNSName.

La confrontation est effectuée en utilisant les règles de correspondance spécifiées par la [RFC2459]. Si plus d'une identité d'un certain type est présente dans le certificat (par exemple, plus d'un nom dNSName) une correspondance avec l'un quelconque de l'ensemble est considéré comme acceptable. Les noms peuvent contenir le caractère générique *(wildcard)* * qui est considéré comme correspondant à tout composant ou fragment de composant d'un seul nom de domaine. Par exemple, *.a.com correspond pour foo.a.com mais pas pour bar.foo.a.com, f*.com correspond pour foo.com mais pas pour bar.com.

Dans certains cas, l'URI est spécifié comme une adresse IP plutôt que comme nom d'hôte. Dans ce cas, le iPAddress subjectAltName doit être présent dans le certificat et doit correspondre exactement à l'IP dans l'URI.

Si le nom d'hôte ne correspond pas à l'identité qui figure dans le certificat, les clients en mode utilisateur DOIVENT soit notifier l'utilisateur (les clients PEUVENT donner à l'utilisateur l'opportunité de continuer la connexion dans tous les cas) soit terminer la connexion avec une erreur "mauvais certificat". Les clients automatisés DOIVENT enregistrer l'erreur sur un journal d'audit approprié (s'il en est un disponible) et DEVRAIENT mettre fin à la connexion (avec une erreur "mauvais

certificat"). Les clients automatisés PEUVENT fournir un réglage de configuration qui désactive cette vérification, mais DOIVENT fournir un réglage qui l'active.

Noter que dans beaucoup de cas l'URI lui-même vient d'une source qui n'est pas de confiance. La vérification décrite ci-dessus ne donne pas de protection contre des attaques où cette source est compromise. Par exemple, si l'URI a été obtenu en cliquant sur une page HTML qui a elle-même été obtenue sans utiliser HTTP/TLS, un attaquant interposé a pu remplacer l'URI. Pour prévenir cette forme d'attaque, les utilisateurs devraient examiner attentivement le certificat présenté par le serveur pour déterminer si il satisfait à leurs attentes.

3.2 Identité du client

Normalement, le serveur n'a pas de connaissance externe de ce que devrait être l'identité du client et par conséquent, aucune vérification n'est possible (autre que celle que le client a une chaîne de certificats dont la racine est une autorité de certification appropriée). Si un serveur a une telle connaissance (normalement, par une source externe à HTTP ou TLS) il DEVRAIT vérifier l'identité comme décrit ci-dessus.

Références

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

[RFC2246] T. Dierks et C. Allen, "Protocole TLS version 1.0", janvier 1999.

[RFC2459] R. Housley, W. Ford, W. Polk et D. Solo, "Profil de certificat d'infrastructure de clé publique X.509 et de CRL pour l'Internet", janvier 1999. (*Obsolète, voir la* RFC5280) *(P.S.)*

[RFC2616] R. Fielding et autres, "Protocole de transfert hypertexte -- HTTP/1.1", juin 1999. *(D.S., MàJ par* 2817, 6585)

[RFC2817] R. Khare, S. Lawrence, "Mise à niveau de TLS au sein de HTTP/1.1", mai 2000. *(P.S.)*

Considérations sur la sécurité

Le présent document est tout entier consacré à la sécurité.

Adresse de l'auteur

Eric Rescorla
RTFM, Inc.
30 Newell Road, #16
East Palo Alto, CA 94303

téléphone : (650) 328-8631
mél : ekr@rtfm.com

Déclaration complète de droits de reproduction

FORCE déclinent toutes garanties, exprimées ou implicites, y compris mais non limitées à toute garantie que l'utilisation des informations ci-encloses ne violent aucun droit ou aucune garantie implicite de commercialisation ou d'aptitude à un objet particulier.

Propriété intellectuelle

L'IETF ne prend pas position sur la validité et la portée de tout droit de propriété intellectuelle ou autres droits qui pourraient être revendiqués au titre de la mise en œuvre ou l'utilisation de la technologie décrite dans le présent document ou sur la mesure dans laquelle toute licence sur de tels droits pourrait être ou n'être pas disponible ; pas plus qu'elle ne prétend avoir accompli aucun effort pour identifier de tels droits. Les informations sur les procédures de l'ISOC au sujet des droits dans les documents de l'ISOC figurent dans les BCP 78 et BCP 79.

Des copies des dépôts d'IPR faites au secrétariat de l'IETF et toutes assurances de disponibilité de licences, ou le résultat de tentatives faites pour obtenir une licence ou permission générale d'utilisation de tels droits de propriété par ceux qui mettent en œuvre ou utilisent la présente spécification peuvent être obtenues sur le répertoire en ligne des IPR de l'IETF à http://www.ietf.org/ipr .

L'IETF invite toute partie intéressée à porter son attention sur tous copyrights, licences ou applications de licence, ou autres droits de propriété qui pourraient couvrir les technologies qui peuvent être nécessaires pour mettre en œuvre la présente norme. Prière d'adresser les informations à l'IETF à ietf- ipr@ietf.org .

Remerciement

Le financement de la fonction d'édition des RFC est actuellement fourni par l'Internet Society.