Équipe d'ingénierie de l'Internet (IETF)
Request for Comments : 9611
Catégorie : Sur la voie de la normalisation
ISSN: 2070-1721
Traduction Claude Brière de L'Isle

A. Antony, secunet
T. Brunner, codelabs
S. Klassert, secunet
P. Wouters, Aiven
juillet 2024

Prise en charge de la version 2 du protocole d'échange de clé Internet (IKEv2) pour les associations de sécurité filles par ressource

Résumé
Afin d'augmenter la bande passante du trafic IPsec entre homologues, le présent document définit une charge utile Types d'état de message de notification et une charge utile Types d'erreur de message de notification pour le protocole IKEv2 *(Internet Key Exchange Protocol Version 2)* afin de prendre en charge la négociation de plusieurs associations de sécurité (SA) filles avec les mêmes sélecteurs de trafic qu'utilisés sur les différentes ressources, telles que les processeurs.

La notification SA_RESOURCE_INFO est utilisée pour porter l'information que l'association de sécurité fille négociée et les nouvelles SA filles ultérieures avec les mêmes sélecteurs de trafic constituent un groupe logique de SA filles où la plupart ou la totalité des SA filles sont liées à une ressource spécifique, telle qu'un processeur spécifique. La notification TS_MAX_QUEUE indique que l'homologue n'est pas disposé à créer plus de SA filles supplémentaires pour cette combinaison particulière de sélecteurs de trafic négociée.

L'utilisation de plusieurs SA filles avec les mêmes sélecteurs de trafic présente l'avantage que chaque ressource contenant la SA fille dispose de son propre compteur de numéros de séquence, ce qui garantit que les processeurs n'ont pas à synchroniser leur état cryptographique ou à désactiver leur protection contre la répétition de paquets.

Statut de ce mémoire
Ceci est un document de l'Internet sur la voie de la 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/rfc9611.

Table des matières

1. Introduction

La plupart des mises en œuvre de IPsec sont actuellement limitées à l'utilisation d'une file d'attente matérielle ou d'une seule ressource de module processeur de commande (CPU, *Control Processor Unit*) pour une SA fille. L'exécution du chiffrement du flux de paquets en parallèle peut être effectuée, mais il existe un goulot d'étranglement entre différentes parties du matériel qui se verrouillent ou attendent d'obtenir le numéro de séquence attribué pour le paquet à chiffrer. Le résultat est qu'une machine avec plusieurs de ces ressources est limitée à l'utilisation d'une seule de ces ressources par SA fille. Cela limite considérablement le débit qui peut être atteint. Par exemple, au moment de la rédaction de ce texte, une liaison non chiffrée de 10 Gbit/s ou plus est généralement réduite à 2-5 Gbit/s lorsque IPsec est utilisé pour chiffrer la liaison à l'aide d'AES-GCM. En utilisant la mise en œuvre spécifiée dans le présent document, le débit agrégé est passé de 5 Gbit/s avec un processeur à 40-60 Gbit/s avec 25-30 CPU.

Bien que cela puisse être (partiellement) atténué par la configuration de plusieurs SA filles plus étroites (par exemple, à l'aide de la fonction PFP *(Populate From Packet)* comme spécifié dans l'architecture IPsec [RFC4301]) cette fonctionnalité IPsec entraînerait trop de SA filles (une par flux du réseau) ou trop peu (un flux réseau utilisé sur plusieurs processeurs). De plus, la fonction PFP n'est pas largement mise en œuvre.

Pour faire un meilleur usage de plusieurs files d'attente réseau et processeurs, il peut être avantageux de négocier et d'installer plusieurs SA filles avec des sélecteurs de trafic identiques. IKEv2 [RFC7296] permet déjà d'installer plusieurs SA filles avec des sélecteurs de trafic identiques, mais il n'offre pas de méthode pour indiquer que la SA fille supplémentaire est demandée pour des raisons d'augmentation des performances et qu'elle est restreinte à une ressource (file d'attente ou CPU).

Lorsqu'un homologue IKEv2 reçoit pour un seul ensemble de sélecteurs de trafic plus de SA filles supplémentaires qu'il n'en veut créer, il peut retourner une notification d'erreur de TS_MAX_QUEUE.

1.1 Notation des exigences

Les mots clés "DOIT", "NE DOIT PAS", "EXIGÉ", "DEVRA", "NE DEVRA PAS", "DEVRAIT", "NE DEVRAIT PAS", "RECOMMANDÉ", "NON RECOMMANDÉ", "PEUT" et "FACULTATIF" dans ce document doivent être interprétés comme décrit dans le BCP 14 [RFC2119] [RFC8174] lorsque, et seulement lorsqu'ils apparaissent en majuscules, comme indiqué ici.

1.2 Terminologie

Le présent document utilise les termes suivants définis dans IKEv2 [RFC7296] : données de notification, sélecteur de trafic (TS, *Traffic Selector*), initiateur de sélecteur de trafic (Tsi, *Traffic Selector initiator*), répondeur de sélecteur de trafic (TSr, *Traffic Selector responder*), SA fille, charge utile de configuration (CP, *Configuration Payload*), SA IKE, CREATE_CHILD_SA, et NO_ADDITIONAL_SAS.

Le présent document utilise aussi les termes suivants définis dans la [RFC4301] : base de données de politique de sécurité (SPD, *Security Policy Database*), association de sécurité (SA, *Security Association*).

2. Goulots d'étranglement des performances

Il existe plusieurs raisons pratiques pour lesquelles la plupart des mises en œuvre doivent limiter une association de sécurité (SA) fille à une seule ressource matérielle spécifique. Une des principales limitations découle des défis associés au partage des états cryptographiques, des compteurs et des numéros de séquence entre plusieurs processeurs. Lorsque ces processeurs tentent d'utiliser simultanément des états partagés, il devient impossible de le faire sans encourir une pénalité de performance significative. Il est nécessaire de négocier et d'établir plusieurs SA filles avec un initiateur de sélecteur de trafic (TSi) et un répondeur de sélecteur de trafic (TSr) identiques pour chaque ressource.

3. Négociation de SA filles spécifiques des ressources

Un échange IKEv2 initial est utilisé pour établir une SA IKE et la SA fille initiale. Si il désire plusieurs SA filles avec les mêmes sélecteurs de trafic liés à une seule ressource, l'initiateur ajoute la charge utile de notification SA_RESOURCE_INFO à la charge utile Exchange qui négocie la SA fille (par exemple, IKE_AUTH ou CREATE_CHILD_SA). Si cette SA fille initiale va être liée à une ressource spécifique, elle PEUT l'indiquer en incluant un identifiant dans les données de notification. Un répondeur qui veut avoir plusieurs SA filles pour les mêmes sélecteurs de trafic répondra en ajoutant également la charge utile de notification SA_RESOURCE_INFO dans laquelle il PEUT ajouter des données de notification non nulles.

Des SA filles supplémentaires spécifiques de ressources sont négociées comme des SA filles ordinaires à l'aide de l'échange CREATE_CHILD_SA et sont identifiées de la même manière par une notification SA_RESOURCE_INFO qui les accompagne.

Lors de l'installation, chaque SA fille spécifique d'une ressource est associée à un sélecteur local supplémentaire, tel que le processeur. Ces SA filles spécifiques d'une ressource DOIVENT être négociées avec des propriétés de SA filles identiques à celles qui ont été négociées pour la SA fille initiale. Cela inclut les algorithmes cryptographiques, les sélecteurs de trafic, le mode (par exemple, de transport) l'utilisation de la compression, etc. Cependant, chaque SA fille a son propre matériel de chiffrement qui est déduit individuellement selon le processus IKEv2 habituel. La charge utile de notification SA_RESOURCE_INFO PEUT être vide ou PEUT contenir des données d'identification. Ces données d'identification DEVRAIENT être un identifiant unique au sein de toutes les SA filles avec les mêmes charges utiles TS, et l'homologue DOIT uniquement les utiliser à des fins de débogage.

Des SA filles supplémentaires peuvent être démarrées à la demande ou en une seule fois. Les homologues peuvent aussi supprimer des SA filles spécifiques par ressource s'ils jugent que la ressource associée est inactive.

Durant le renouvellement de clés CREATE_CHILD_SA pour la SA fille, la notification SA_RESOURCE_INFO PEUT être incluse, mais qu'elle soit incluse ou non, la SA fille dont la clé est renouvelée devrait être liée à la ou aux mêmes ressources que la SA fille qui est en cours de renouvellement de clés.

4. Considérations de mise en œuvre

Il y a diverses considérations qu'une mise en œuvre peut utiliser pour déterminer la meilleure procédure pour installer plusieurs SA filles.

Une procédure simple pourrait être d'installer une SA fille supplémentaire sur chaque CPU. Une mise en œuvre peut s'assurer qu'une SA fille peut être utilisée par tous les processeurs, de sorte que lors de la négociation d'une nouvelle SA fille par CPU, qui prend généralement un délai d'aller-retour, le CPU sans SA fille spécifique du processeur peut toujours chiffrer ses paquets à l'aide de la SA fille disponible pour tous les CPU. Autrement, si une mise en œuvre trouve qu'elle a besoin de chiffrer un paquet mais que le processeur actuel n'a pas les ressources pour chiffrer ce paquet, elle peut relayer ce paquet à un processeur spécifique qui a la capacité de chiffrer le paquet, bien que cela se fasse au prix des performances.

L'exécution de négociations de SA fille par processeur peut entraîner le lancement simultané de SA filles supplémentaires par les deux homologues. Cela est particulièrement probable si les SA filles par processeur sont déclenchées par des messages SADB_ACQUIRE individuels [RFC2367]. Les répondeurs devraient installer la SA fille supplémentaire sur un CPU qui a le moins de SA filles supplémentaires pour cette paire de TSi/TSr.

Quand le nombre de ressources de file d'attente ou de processeur est différent entre les homologues, l'homologue qui a le moins de ressources peut décider de ne pas installer de deuxième SA fille sortante pour la même ressource, car il ne l'utilisera jamais pour envoyer du trafic. Cependant, il doit installer toutes les SA filles entrantes, car il s'est engagé à recevoir du trafic sur ces SA filles négociées.

Si des messages de déclenchement de paquet par CPU (par exemple, SADB_ACQUIRE) sont mis en œuvre (voir à la Section 6) l'entrée d'initiateur de sélecteur de trafic (Tsi, *Traffic Selector initiator*) contenant les informations du paquet déclencheur doit être incluse dans l'ensemble TS de la même manière que les SA filles normales, comme spécifié au paragraphe 2.9 de la [RFC7296]. Sur la base de l'entrée TSi déclencheuse, une mise en œuvre peut choisir le processeur cible le plus optimal pour y installer la SA fille supplémentaire. Par exemple, si le paquet déclencheur était pour une destination TCP à l'accès 25 (SMTP) elle peut être en mesure d'installer la SA fille sur le CPU qui exécute également le processus du serveur de messagerie. Les sélecteurs de trafic de paquets de déclenchement sont documentés dans IKEv2 [RFC7296], paragraphe 2.9.

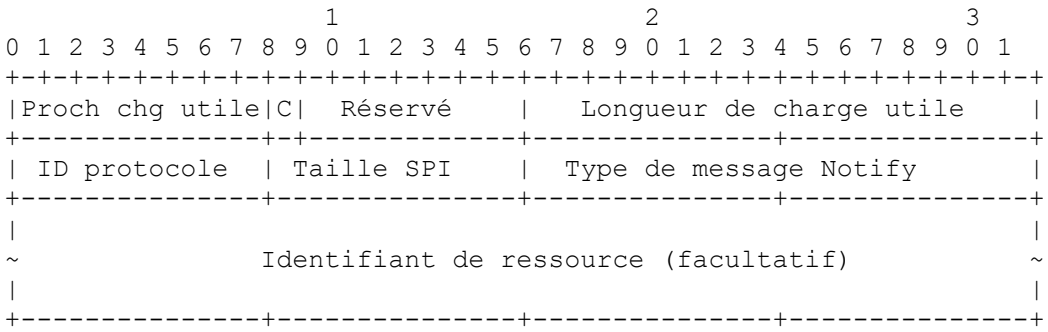
Selon IKEv2, le changement de clés d'une SA fille DEVRAIT utiliser les mêmes (ou plus larges) sélecteurs de trafic pour s'assurer que la nouvelle SA fille couvre tout ce que couvre la SA fille dont les clés ont changé. Cela inclut les sélecteurs de trafic négociés via des charges utiles de configuration telles que INTERNAL_IP4_ADDRESS, qui peuvent utiliser le large ensemble de TS d'origine ou utiliser l'ensemble de TS restreint.

5. Format de charge utile

Le format de charge utile Notify est défini au paragraphe 3.10 de IKEv2 [RFC7296], et est copié ici dans un souci pratique.

Tous les champs multi-octets représentant des entiers sont affichés en ordre gros boutien (aussi appelé "octet de poids fort en premier", ou "ordre des octets du réseau").

5.1 Charge utile de type d'état de message Notify SA_RESOURCE_INFO



Fanion C (critique) - DOIT être 0.

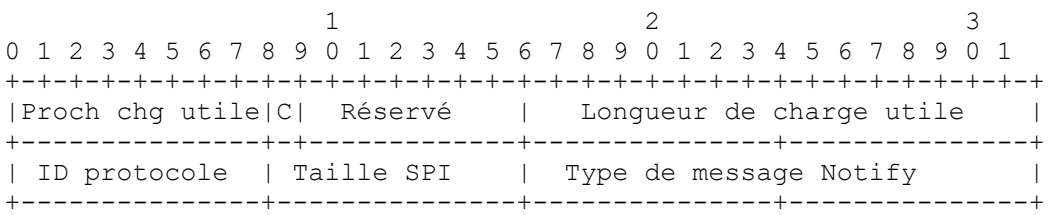
ID de protocole (1 octet) - DOIT être 0, DOIT être ignoré sinon

Taille de SPI (1 octet) - DOIT être 0, DOIT être ignoré sinon.

Valeur du type de message d'état Notify (2 octets) – réglée à 16444.

Identifiant de ressource (facultatif) – ces données opaques peuvent être réglée à transporter l'identité locale de la ressource.

5.2 Charge utile de type d'erreur de message Notify TS_MAX_QUEUE



Fanion C (critique) - DOIT être 0.

ID de protocole (1 octet) - DOIT être 0, DOIT être ignoré sinon

Taille de SPI (1 octet) - DOIT être 0, DOIT être ignoré sinon.

Type d'erreur de message Notify (2 octets) – réglé à 48.

Il n'y a pas de données associées à ce type Notify.

6. Considérations de fonctionnement

Les mises en œuvre qui prennent en charge les SA par CPU DEVRAIENT étendre leur sélecteur SPD local, et le mécanisme de négociation à la demande qui est déclenché par le trafic, pour inclure un identifiant de CPU (ou de file d'attente) dans leur message de déclenchement de paquets (par exemple, SADB_ACQUIRE) du SPD à l'automate IKE. Une mise en œuvre qui ne prend pas en charge la réception de messages de déclenchement de paquets par processeur PEUT initier toutes ses SA filles immédiatement après la réception du (seul) message de déclenchement de paquets qu'elle recevra de la pile IPsec. Une telle mise en œuvre doit également être prudente lors de la réception d'une demande Delete Notify pour une SA fille par CPU, car elle n'a pas de méthode pour détecter quand elle doit réactiver plus tard une telle SA fille par CPU. Aussi, le fait de réactiver l'association SA fille par CPU supprimée immédiatement après avoir reçu la notification de suppression peut provoquer une boucle infinie entre les homologues. Un autre problème lié au fait de ne pas activer toutes ses SA filles par processeur est que si l'homologue agit de la même manière, les deux homologues peuvent se retrouver avec uniquement la première SA fille sans jamais activer de SA fille par CPU. Il est donc RECOMMANDÉ de mettre en œuvre des messages de déclenchement de paquets par CPU.

Les homologues DEVRAIENT être souples avec le nombre maximum de SA filles qu'ils autorisent pour une combinaison TSi/TSr donnée afin de tenir compte des cas limites. Par exemple, lors du changement de clés de la SA fille, un grand nombre de SA filles supplémentaires pourraient être créées avant que les anciennes SA filles soient supprimées. De même, lors de l'utilisation d'associations SA filles à la demande, les deux extrémités pourraient déclencher plusieurs demandes de SA filles, car le paquet initial à l'origine de la négociation de SA fille aurait pu être transporté à l'homologue via la première SA fille, où son paquet de réponse pourrait également déclencher le démarrage d'une négociation de SA fille à la demande. Comme les SA filles supplémentaires consomment peu de ressources supplémentaires, il est RECOMMANDÉ de permettre au moins le double du nombre de CPU disponibles. Une mise en œuvre PEUT autoriser un nombre illimité de SA filles supplémentaires et ne limiter ce nombre qu'en fonction de ses stratégies génériques de protection des ressources qui sont utilisées pour exiger des COOKIES ou refuser de nouvelles négociations IKE ou de SA filles. Bien qu'avoir un très grand nombre (par exemple, des centaines ou des milliers) de SA puisse ralentir la recherche dans une base de données d'association de sécurité (SAD, *Security Association Database*) par paquet.

Les mises en œuvre peuvent prendre en charge le déplacement dynamique d'une SA fille par processeur d'un CPU à un autre. Si cette méthode est prise en charge, les mises en œuvre doivent veiller à déplacer à la fois les SA entrantes et sortantes. Si le point de terminaison IPsec est une passerelle, il peut déplacer les SA entrantes et sortantes indépendamment l'une de l'autre. Il est probable que pour une passerelle, le trafic IPsec soit asymétrique. Si le point de terminaison IPsec est aussi l'hôte responsable de la génération du trafic, les SA entrantes et sortantes DEVRAIENT rester comme une paire sur le même CPU. Si un hôte a précédemment ignoré l'installation d'une SA sortante parce qu'il s'agirait d'une SA sortante dupliquée inutilisée, il devra créer et ajouter la SA sortante précédemment ignorée à la SAD avec le nouvel ID de processeur. Il se peut que la SA entrante n'ait pas d'identifiant de processeur dans la SAD. L'ajout de la SA sortante à la SAD nécessite l'accès au matériel de chiffrement, tandis que la mise à jour du sélecteur de processeur sur des SA sortantes existantes peut ne pas nécessiter l'accès au matériel de chiffrement. Pour prendre en charge cela, le logiciel IKE peut devoir conserver le matériel de chifrement plus longtemps qu'il ne le ferait normalement, car il peut tenter activement de détruire le matériel de chiffrement de la mémoire auquel l'automate IKE n'a plus besoin d'accéder.

Une mise en œuvre qui n'accepte pas d'autres SA filles spécifiques d'une ressource NE DOIT PAS retourner l'erreur NO_ADDITIONAL_SAS, car elle pourrait être mal interprétée par l'homologue comme signifiant qu'aucune autre SA fille avec un TSi et/ou TSr différent n'est non plus autorisée. Au lieu de cela, il DOIT retourner TS_MAX_QUEUE.

7. Considérations sur la sécurité

De la même manière qu'une mise en œuvre devrait limiter le nombre de SA semi-ouvertes pour limiter l'impact d'une attaque de déni de service, il est RECOMMANDÉ qu'une mise en œuvre limite le nombre maximum de SA filles supplémentaires autorisées par un TSi/TSr.

L'utilisation de plusieurs SA filles spécifiques d'une ressource a du sens pour les connexions IPsec à volume élevé sur des machines de passerelle IPsec où l'administrateur a une relation de confiance avec l'administrateur de l'homologue et où l'abus est peu probable et facilement résolu.

Cette relation de confiance n'est généralement pas présente pour les déploiements de VPN à accès distant, et l'autorisation de SA filles par processeur n'est PAS RECOMMANDÉE dans ces scénarios. Donc, il est aussi NON RECOMMANDÉ d'autoriser par défaut les SA filles par CPU.

La notification SA_RESOURCE_INFO contient une charge utile de données facultative qui peut être utilisée par

l'homologue pour identifier la SA fille appartenant à une ressource spécifique. Les données de notification NE DEVRAIENT PAS être un identifiant qui puisse être utilisé pour obtenir des informations sur le matériel. Par exemple, l'utilisation du numéro du CPU lui-même comme identifiant peut permettre à un attaquant de savoir quels paquets sont traités par quel identifiant de CPU et d'optimiser une attaque en force brute contre le système.

8. Considérations relatives à l'IANA

L'IANA a enregistré une nouvelle valeur dans le registre "Types d'état de message Notify IKEv2".

Valeur	Types d'état de message Notify	Référence
16444	SA_RESOURCE_INFO	RFC 9611

Tableau 1

L'IANA a enregistré une nouvelle valeur dans le registre "Types d'erreur de message Notify IKEv2".

Valeur	Types d'erreur de message Notify	Référence
48	TS_MAX_QUEUE	RFC 9611

Tableau 2

9. Références

9.1 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). DOI 10.17487/RFC2119.

[RFC7296] C. Kaufman, et autres, "Protocole d'échange de clé Internet version 2 (IKEv2)", octobre 2014. STD 79. DOI 10.17487/RFC7296, *(MàJ par* RFC7670, RFC8247).

[RFC8174] B. Leiba, "Ambiguïté des mots clés en majuscules ou minuscules dans la RFC2119", mai 2017. BCP14. *(MàJ* 2119). DOI 10.17487/RFC8174.

9.2 Références pour information

[RFC2367] D. McDonald, C. Metz, B. Phan, "API de gestion de clé PF_KEY, version 2", juillet 1998. *(Info.)* DOI 10.17487/RFC2367.

[RFC4301] S. Kent et K. Seo, "Architecture de sécurité pour le protocole Internet", décembre 2005. DOI 10.17487/RFC4301, *(P.S. ; remplace la* RFC2401)

Remerciements

Les personnes suivantes ont assuré la relecture et ont fourni de précieux retours : Roman Danyliw, Warren Kumari, Tero Kivinen, Murray Kucherawy, John Scudder, Valery Smyslov, Gunter van de Velde, et Ã‰ric Vyncke.

Adresse des auteurs

Antony Antony
secunet Security Networks AG
mél : antony.antony@secunet.com

Tobias Brunner
codelabs GmbH
mél : tobias@codelabs.ch

Steffen Klassert
secunet Security Networks AG
mél : steffen.klassert@secunet.com

Paul Wouters
Aiven
mél : paul.wouters@aiven.io