Groupe de travail Réseau N.Williams, Sun
Request for Comments: 5660
Catégorie : Sur la voie de la normalisation octobre 2009
Traduction Claude Brière de L'Isle

Canaux IPsec : verrouillage de connexion

Résumé
Le présent document spécifie, de façon abstraite, comment les protocoles d'applications et de transport s'interfacent avec IPsec afin de créer des "canaux" en verrouillant les "connexions" (flux de paquets) pour certains paramètres d'associations de sécurité (SA, *Security Association*) IPsec pour la durée de vie des connexions. Le verrouillage de connexion est mis en couche par dessus IPsec et ne modifie pas l'architecture IPsec sous-jacente.

Le verrouillage de connexion peut être utilisé pour protéger les applications contre une exposition accidentelle de flux de paquets à des homologues non voulus, que ce soit par suite d'une reconfiguration de IPsec ou par suite de l'utilisation d'une identité d'homologue faible pour les associations d'adresses d'homologues. Une association faible de l'identifiant d'homologue et des adresses d'homologue est au cœur de la sécurité mieux que rien (BTNS, *Better Than Nothing Security*) ; donc, le verrouillage de connexion peut ajouter une mesure de protection significative aux nœuds IPsec BTNS.

Finalement, la disponibilité des canaux IPsec va rendre possible d'utiliser les liens de canaux aux canaux IPsec.

Table des matières

1. Introduction

IPsec protège les paquets avec peu ou pas d'égards aux flux de paquets à états pleins associés à des protocoles de couche supérieure (ULP, *upper-layer protocol*). Cela expose les applications qui s'appuient sur IPsec pour la protection de session aux risques associés au changement des configurations IPsec, aux configurations qui permettent que plusieurs homologues accèdent aux mêmes adresses, et/ou à une association faible des identifiants d'homologue et de leurs adresses. Ce dernier risque peut se produire par suite d'une correspondance "de caractère générique" dans la base de données d'autorisation d'homologue (PAD, *Peer Authorization Database*) IPsec, en particulier quand la "sécurité mieux que rien" (BTNS, *Better Than Nothing Security*) [RFC5387] est utilisée.

Les applications qui souhaitent utiliser IPsec peuvent devoir s'assurer que la politique locale sur les divers points d'extrémité est configurée de façon appropriée [RFC5406], [USING-IPSEC]. Il n'y a pas d'interfaces de programmation d'application (API, *Application Programming Interface*) standard pour faire cela (bien qu'il y ait des API non standard, comme [IP_SEC_OPT]) -- dont une conséquence majeure, par exemple, est que les applications doivent encore utiliser les noms d'hôte (et, par exemple, le système des noms de domaine [RFC1034]) et les adresses IP dans les API existantes et doivent dépendre d'une configuration IPsec qu'elles peuvent n'être pas capables de vérifier. En plus de spécifier les aspects de configuration exigée de la base de données de politique de sécurité (SPD, *Security Policy Database*) les spécifications d'application doivent aussi traiter la configuration de PAD/SPD pour lier fortement les adresses individuelles aux identités IPsec individuelles et aux accréditifs (certificats, clés publiques, etc.).

L'utilisation de IPsec est alors assez bizarre pour les applications. Pour traiter cela, on a besoin d'API pour IPsec. Pas simplement d'API pour configurer IPsec, mais aussi des API similaires aux API IP existantes (par exemple, des "prises BSD") afin que les applications typiques qui utilisent UDP [RFC0768], TCP [RFC0793], et le protocole de transmission de commandes de flux (SCTP, *Stream Control Transmission Protocol*) [RFC4960] puissent utiliser IPsec avec un minimum de changements.

Le présent document décrit les fondements des API IPsec que les applications UDP et TCP peuvent utiliser : une façon de lier les flux de trafic pour, par exemple, des connexions TCP pour les propriétés de sécurité désirées par l'application. On appelle cela des "verrouillages de connexion" (et, dans certains contextes, des "canaux IPsec"). Les méthodes mentionnées ci-dessous réalisent cela en créant une interface des ULP et des applications avec IPsec.

Si il est largement adopté, le verrouillage de connexion pourrait rendre l'utilisation de IPsec par l'application beaucoup plus simple, au moins pour certaines classes d'applications.

Le verrouillage de connexion, comme il est spécifié ici, est principalement sur la surveillance de mises à jour à la SPD et à la base de données d'association de sécurité (SAD, *Security Association Database*) pour détecter les changements qui sont contraires aux exigences d'une application pour tout flux de paquets donné, et pour réagir en conséquence (comme par une alerte synchronisée de l'ULP et de l'application avant que des paquets puissent être envoyés ou reçus selon la nouvelle politique). En aucun cas les bases de données de politique IPsec ne sont modifiées par le verrouillage de connexion de quelque façon que ce soit qui puisse persister au delà de la durée de vie des flux de paquets concernés, ni des réamorçages. Dans aucun cas la PAD n'est à modifier par le verrouillage de connexion. Si toutes les caractéristiques facultatives du verrouillage de connexion sont exclues, alors le verrouillage de connexion peut être mis en œuvre comme un surveillant de SPD et des changements de SAD qui surgissent dans leurs travaux pas plus que ce qui est nécessaire pour fournir des alertes synchrones aux ULP et applications.

On suppose que le lecteur est familier de l'architecture IPsec [RFC4301] et de la version 2 du protocole d'échange de clé Internet (IKEv2, *Internet Key Exchange Protocol version 2*) [RFC4306].

Note : les termes "verrouillage de connexion" et "canal IPsec" sont utilisés de façon interchangeable ci-dessous. Ce dernier terme se rapporte au "lien de canal" [RFC5056]. Le verrouillage de connexion convient pour l'usage dans les applications de lien de canal, ou le sera, à tout le moins, quand les liens de canal pour les canaux IPsec seront définis

(la spécification des liens de canal IPsec sort du domaine d'application de ce document).

Note : lorsque le présent document mentionne un identifiant d'homologue IPsec, il se réfère à l'identifiant d'homologue de l'échange de clé Internet (IKE, *Internet Key Exchange*) (par exemple, l'identifiant déduit du certificat d'un homologue, ainsi que le certificat) pas à l'adresse IP de l'homologue.

1.1 Conventions utilisées dans le document

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].

Les noms de fonction abstraite sont tout en majuscules et notés par une paire de parenthèses. Dans leurs descriptions, les arguments apparaissent au sein de parenthèses, avec les arguments facultatifs entourés de crochets. Les valeurs retournées, si il en est, sont indiquées en faisant suivre la liste d'arguments de fonction par "->" et une description de la valeur retournée. Par exemple, "FOO(3-tuple, [message])" serait une fonction nommée "FOO" avec deux arguments, dont un facultatif, et ne retournant rien, tandis que "FOOBAR(bride) -> état" serait une fonction avec un seul argument exigé qui retourne une valeur. Les types de valeurs sont décrits dans le texte environnant.

2. Verrouillage de connexion

Un "canal IPsec" est un flux de paquets associé à un bloc de contrôle d'ULP, comme une connexion TCP, où tous les paquets sont protégés par des SA IPsec telles que :
- o l'identité de l'homologue est la même pour la durée de vie du flux de paquets ;
- o la qualité de la protection IPsec utilisée pour les paquets individuels du flux de paquets est la même pour tous pour la durée de vie du flux de paquets.

Un canal IPsec est créé quand le flux de paquets associé est créé. Ce peut être le résultat d'une opération locale (par exemple, un connect()) qui cause l'envoi du paquet sortant initial pour ce flux, ou ce peut être le résultat de la réception du premier paquet ou du paquet initiateur de ce flux (par exemple, un paquet TCP SYN).

Un canal IPsec est détruit quand le flux de paquets associé se termine. Un canal IPsec peut aussi être "cassé" quand le verrouillage de connexion ne peut pas être maintenu pour une raison quelconque (voir ci-dessous) et dans ce cas l'ULP et l'application en sont informés.

Les canaux IPsec sont créés en "verrouillant" divers paramètres mentionnés ci-dessous sur une connexion d'ULP quand les connexions sont créées. L'ensemble EXIGÉ de paramètres liés dans les canaux IPsec est :
- o le type de protection : confidentialité et/ou protection de l'intégrité ;
- o mode transport contre mode tunnel ;
- o qualité de protection (QoP) : suites d'algorithmes cryptographiques, longueurs de clé, et protection contre la répétition (voir le paragraphe 2.1) ;
- o identité locale : l'identifiant local attribué à l'homologue, selon le modèle de traitement IPsec [RFC4301] et la BTNS [RFC5386] ;
- o identité de l'homologue : identifiants affirmés et autorisés de l'homologue, selon le modèle de traitement IPsec [RFC4301] et la BTNS [RFC5386].

Les SA qui protègent les paquets d'un canal IPsec donné peuvent changer au fil du temps en ce qu'elles peuvent arriver à expiration et être remplacées par des SA équivalentes, ou leurs clés peuvent être changées. L'ensemble de SA qui protègent les paquets d'un canal IPsec n'a pas besoin d'être en rapport avec autre chose que le fait qu'elles doivent être congruentes au canal (c'est-à-dire, les paramètres des SA doivent correspondre à ceux qui sont verrouillés dans le canal). En particulier, il est souhaitable que les canaux IPsec survivent à l'expiration des SA IKE et des SA filles parce que les considérations de fonctionnement des divers protocoles d'échange de clé ne peuvent alors pas affecter la conception et les caractéristiques de verrouillage de connexion.

Quand se produit une situation dans laquelle la SPD est modifiée, ou qu'une SA est ajoutée à la SAD, comme quand la nouvelle politique et/ou la SA ne sont pas congruentes à un canal établi (voir le paragraphe précédent) on considère alors qu'il y a conflit. La résolution de conflit est traitée ci-dessous.

Exigences et recommandations :

- Si un canal IPsec est désiré, alors les paquets pour une connexion donnée NE DOIVENT PAS être envoyés tant que le canal n'est pas établi.

- Si un canal IPsec est désiré, alors des paquets entrants pour une connexion donnée NE DOIVENT PAS être acceptés tant que le canal n'est pas établi. C'est-à-dire, les paquets entrants pour une connexion donnée qui arrivent avant l'établissement du canal IPsec correspondant doivent être éliminés ou l'établissement du canal doit échouer.

- Une fois qu'un canal IPsec est établi, les paquets pour la connexion verrouillée NE DOIVENT PAS être envoyés non protégés ni protégés par une SA qui ne correspond pas aux paramètres verrouillés.

- Une fois qu'un canal IPsec est établi, les paquets pour la connexion verrouillée NE DOIVENT PAS être acceptés non protégés ni protégés par une SA qui ne correspond pas aux paramètres verrouillés. C'est-à-dire, de tels paquets doivent être soit éliminés, soit causer la terminaison du canal, et l'application doit être informée avant que les données provenant d'un tel paquet puissent être livrées à l'application.

- Les mises en œuvre DEVRAIENT fournir des interfaces de programmation pour s'enquérir des valeurs des paramètres verrouillés dans une connexion.

- Les mises en œuvre qui fournissent de telles interfaces de programmation DOIVENT rendre disponibles aux applications toutes les informations pertinentes et disponibles sur l'identifiant d'un homologue, y compris les informations d'authentification. Cela inclut le certificat de l'homologue, quand il en utilise, et l'ancre de confiance à laquelle il a été validé (mais pas nécessairement toute la chaîne de validation de certificat).

- Les mises en œuvre qui fournissent de telles interfaces de programmation DEVRAIENT rendre disponibles aux applications toutes les informations sur les adresses IP publiques et privées locales et/ou distantes, dans le cas de traversée de NAT.

- Les mises en œuvre qui fournissent de telles interfaces de programmation DEVRAIENT rendre disponibles aux applications les adresses internes et externes locales et d'homologue chaque fois que la connexion verrouillée utilise des SA en mode tunnel.

- Les mises en œuvre DEVRAIENT fournir des interfaces de programmation pour régler les valeurs des paramètres à verrouiller dans une connexion qui va être initiée ou acceptée, mais ces interfaces DOIVENT limiter quelles valeurs les applications peuvent demander en accord avec la politique du système (c'est-à-dire, la PAD et SPD IPsec) et les privilèges locaux de l'application. (La politique normale de système peut ne permettre aucun choix aux applications. Des extensions de politique permettant une protection facultative sont décrites au paragraphe 3.1.)

- Les mises en œuvre DEVRAIENT créer automatiquement des canaux IPsec par défaut quand l'application ne demande pas explicitement un canal IPsec. Les mises en œuvre PEUVENT fournir un moyen de désactiver la création automatique de verrouillages de connexion.

- Les paramètres verrouillés dans un canal IPsec DOIVENT rester inchangés une fois le canal établi.

- Les temporisations lors de l'établissement de SA filles avec des paramètres qui correspondent à ceux verrouillés dans un canal IPsec DOIVENT être traitées comme des pertes de paquet (comme cela arrive, par exemple, quand un réseau subit une partition) ; le traitement normal de temporisation d'ULP et/ou d'application et les considérations de retransmission s'appliquent.

- Les mises en œuvre qui ont un processus de gestion de clé redémarrable (ou un "démon") DOIVENT s'arranger pour que les connexions verrouillées existantes soient rompues et déconnectées, ou pour qu'elles survivent au redémarrage du processus d'échange de clés. (Ceci est impliqué par les exigences ci-dessus.) Par exemple, si une telle mise en œuvre compte sur la conservation de certains aspects de l'état de verrouillage de connexion dans le processus de gestion de clé redémarrable (par exemple, des valeurs qui ont potentiellement de grandes représentations, comme les identifiants d'homologue de BTNS) alors soit cet état doit être restauré au redémarrage d'un tel processus, soit les verrouillages de connexion en instance doivent être passés à l'état CLOSED.

- Une politique IPsec dynamique (voir le paragraphe 3.1) relative aux verrouillages de connexion, si il en est, DOIT être supprimée quand les connections verrouillées sont supprimées, et NE DOIT PAS survivre aux réamorçages.

- Quand la détection d'homologue IKE mort (DPD, *dead-peer detection*) conclut que l'homologue distant est mort ou s'est réamorcé, le système DEVRAIT alors considérer que tous les verrouillages de connexion avec cet homologue sont irrémédiablement rompus.

On décrit deux modèles, dont l'un est normatif, de canaux IPsec pour une mise en œuvre IPsec native. Le modèle normatif se fonde sur des interfaces de programmation abstraites sous la forme d'invocations de fonctions entre les ULP et le composant de gestion de clé de IPsec (fondamentalement, la SAD, augmentée d'une base de données de verrouillage (LD, *Latch Database*)). Le second modèle se fonde sur des interfaces de programmation abstraites entre les ULP et la couche d'encapsulation de couche de sécurité/en-tête d'authentification (ESP/AH, *Encapsulating Security Payload / Authentication Header*) IPsec sous la forme d'un étiquetage de méta-données des paquets au sein de la pile IP.

Les deux modèles donnés ci-dessous ne sont cependant pas entièrement équivalents. Un modèle ne peut pas être mis en œuvre avec des cartes d'interface réseau (NIC, *Network Interface card*) qui téléchargent ESP/AH mais n'étiquettent pas les paquets entrants passé au processeur hôte avec les informations de SA, ni ne permettent au processeur hôte d'étiqueter les paquets sortants. Ce même modèle peut être facilement étendu pour prendre en charge le verrouillage de connexion avec des "prises" de datagramme non connectées, alors que l'autre modèle est lié de façon rigide à une notion de "connexions" et ne peut pas être étendu de cette façon. Il peut y avoir d'autres différences mineures entre les deux modèles. Plutôt que de chercher à établir une équivalence pour un ensemble de garanties de sécurité, on a plutôt choisi un modèle comme étant normatif.

On fournit aussi un modèle pour les mises en œuvre non natives, comme les mises en œuvre prises dans la pile (BITS, *bump-in-the-stack*) et de passerelle de sécurité (SG, *Security Gateway*). Le modèle de verrouillage de connexion pour les mises en œuvre non natives n'est pas entièrement caractérisé parce que il dépend de l'estimation de l'état du flux de paquets, qui n'est pas toujours possible. On ne peut pas non plus attendre des mises en œuvre non natives IPsec qu'elles fournissent des API relatives au verrouillage de connexion (les mises en œuvre qui le font pourraient être dites natives). À ce titre, ce troisième modèle ne convient pas pour les applications de lien de canal [RFC5056].

2.1 Verrouillage des paramètres de qualité de protection

Dans IPsec, l'hypothèse des rôles d'initiateur/répondeur IKE n'est pas déterministe. C'est-à-dire, parfois une SA IKE et des SA filles vont être initiées par le "client" (par exemple, l'appelant de la fonction de prise BSD connect()) et parfois par le "serveur" (par exemple, l'appelant de la fonction de prise BSD accept()). Cela signifie que la négociation de la qualité de protection est aussi non déterministe sauf si un des homologues offre une seule suite de chiffrement dans la négociation IKE.

Quand on crée des SA filles étroites avec des sélecteurs de trafic qui correspondent au quintuplet du verrouillage de connexion, il est possible de contraindre la négociation IKE de qualité de protection à une seule suite de chiffrement. Donc, les mises en œuvre DEVRAIENT fournir une API pour demander l'utilisation de telles SA filles. Les mises en œuvre DEVRAIENT considérer une demande d'application pour une QoP spécifique comme impliquant une demande de SA filles étroites.

Quand on utilise des SA avec des sélecteurs de trafic qui comportent plus d'un seul flux, le système peut alors seulement être capable de verrouiller un ensemble de suites de chiffrement, plutôt qu'une seule suite de chiffrement. Dans ce cas, une mise en œuvre DOIT rapporter la QoP utilisée comme indéterminée.

2.2 Automate à états de verrouillage de connexion

Des verrouillages de connexion peuvent exister dans un des quatre états suivants :
- LISTENER *(écoutant)*
- ESTABLISHED *(établi)*
- BROKEN *(cassé)* (il existe des SA qui sont en conflit avec le verrouillage de connexion donné, des changements de SPD en conflit ont été faits, ou DPD a été déclenché et l'homologue est considéré comme mort ou redémarré)
- CLOSED *(clos)* (par l'ULP, l'application ou administrativement)

et toujours avoir un propriétaire, ou détenteur, associé, comme un bloc de contrôle de transmission (TCB, *Transmission Control Block*) d'ULP.

Un verrouillage de connexion peut être né dans l'état LISTENER, qui peut transiter seulement à l'état CLOSED. L'état LISTENER correspond à l'état LISTEN de TCP (et autres ULP) et est associé à des triplets IP, et peut donner lieu à de nouveaux verrouillages de connexion dans l'état ESTABLISHED.

Un verrouillage de connexion peut aussi être né dans les états ESTABLISHED et BROKEN, soit par l'initiative directe d'un ULP, soit quand un événement se produit qui cause la création par un verrouillage LISTENER d'un nouveau verrouillage (dans l'état ESTABLISHED ou BROKEN). Ces états représentent un verrouillage de connexion actif pour un quintuplet de flux de trafic. Les verrouillages de connexion dans ces deux états peuvent transiter dans les deux autres états, ainsi qu'à l'état CLOSED.

Les verrouillages de connexion restent dans l'état CLOSED jusqu'à ce que leurs possesseurs soient informés, sauf lorsque le possesseur a causé la transition, et dans ce cas cet état est passager. Les transitions de l'état ESTABLISHED ou BROKEN à l'état CLOSED devraient normalement être initiées par les possesseurs de verrou, mais les mises en œuvre DEVRAIENT fournir des interfaces administratives à travers lesquelles elles closent les verrouillages actifs.

Les verrouillages de connexion transitent à l'état BROKEN quand il existe des SA dans la SAD dont les sélecteurs de trafic englobent le quintuplet lié par le verrou, et dont l'homologue et/ou les paramètres sont en conflit avec ceux liés par le verrou. Les transitions à l'état BROKEN ont aussi lieu quand surviennent des changements de la SPD qui vont causer l'envoi ou la réception des paquets de la connexion verrouillée avec des paramètres de protection différents de ceux qui ont été verrouillés. Les transitions à l'état BROKEN sont aussi permises quand le DPD IKEv2 conclut que l'homologue distant est mort ou a réamorcé. Les transitions à l'état BROKEN causent toujours une information du possesseur associé. Les verrouillages de connexion dans l'état BROKEN reviennent à l'état ESTABLISHED quand tous les conflits de SA et/ou de SPD sont supprimés.

La plupart des transitions d'état sont le résultat d'actions locales des possesseurs de verrou (des ULP). Les seules exceptions sont : la naissance dans l'état ESTABLISHED provenant de verrouillages dans l'état LISTENER, des transitions de l'état BROKEN, des transitions de l'état BROKEN à l'état ESTABLISHED, et des transitions administratives à l'état CLOSED. (De plus, voir la note de mise en œuvre sur les processus de gestion de clé redémarrables à la Section 2.)

Le diagramme d'état ci-dessous utilise les conventions décrites au paragraphe 1.1 et les événements de transition d'état décrits au paragraphe 2.3.

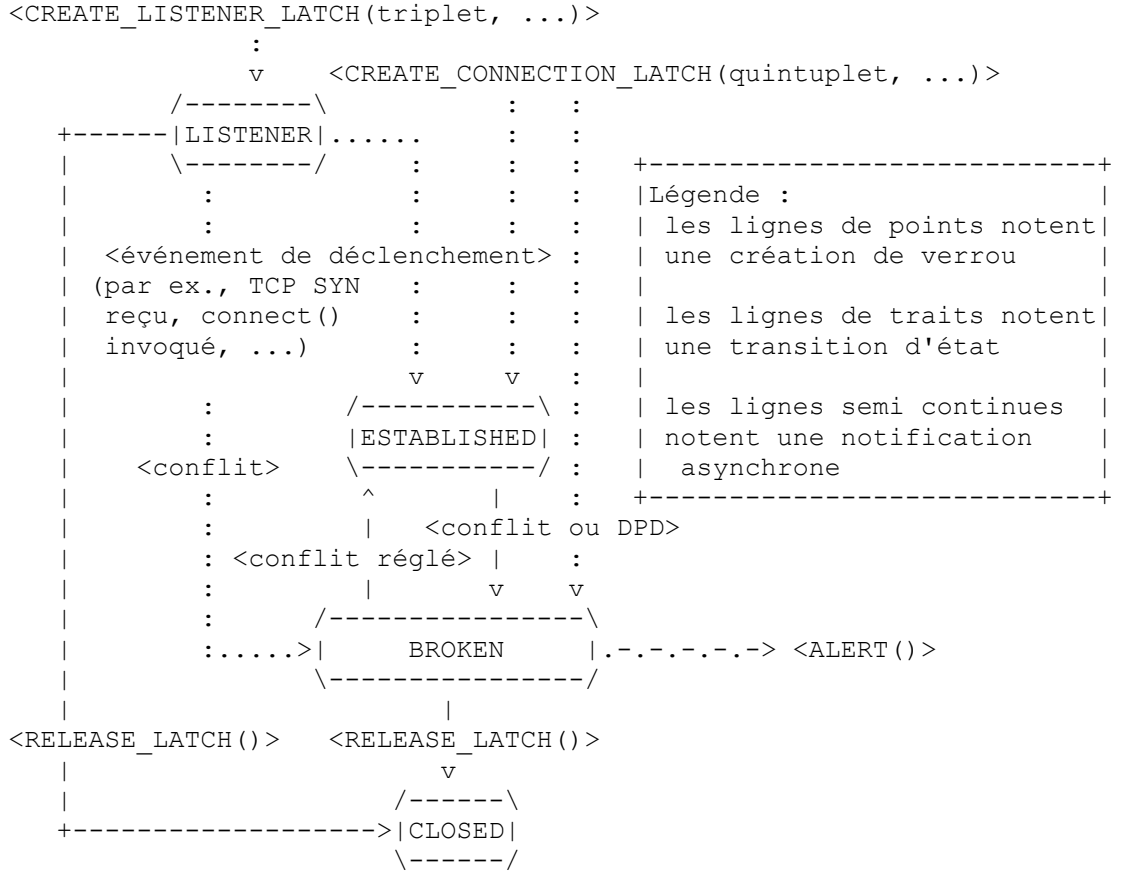

Figure 1 : Automate à états de verrouillage de connexion

Les détails des transitions dépendent du modèle de verrouillage de connexion suivi par une mise en œuvre. Voir les paragraphes suivants.

2.3 Modèle normatif : interfaces d'ULP au gestionnaire de clés

Ce paragraphe décrit le modèle NORMATIF de verrouillage de connexion. Dans ce paragraphe, on décrit le verrouillage de connexion en termes d'interface d'invocation de fonction entre les ULP et le composant "gestionnaire de clé" d'une mise en œuvre native IPsec. Les interfaces abstraites pour créer, s'enquérir de, et libérer les canaux IPsec sont décrites. Ce modèle ajoute un service au gestionnaire de clé IPsec (c'est-à-dire, le composant qui gère la SAD et les interfaces avec une mise en œuvre séparée, ou mise en œuvre directe, des protocoles d'échange de clé) : la gestion des verrouillages de connexion. Il y a a aussi une nouvelle base de données IPsec, la base de données de verrous (LD) qui contient tous les objets de verrouillage de connexion. La LD ne persiste pas à travers les réamorçages de système.

Le modèle traditionnel de traitement IPsec permet l'existence concurrente de SA avec des homologues différents mais des sélecteurs de trafic en chevauchement. Un tel comportement, dans ce modèle, viole directement les exigences du verrouillage de connexion (voir la Section 2). On règle ce problème en exigeant que les verrouillages de connexion soient rompus (et les détenteurs informés) quand de tels conflits surviennent.

La figure pour information suivante illustre ce modèle et l'API dans des termes qui sont familiers à de nombreuses mises en œuvre, bien que non applicables à toutes :

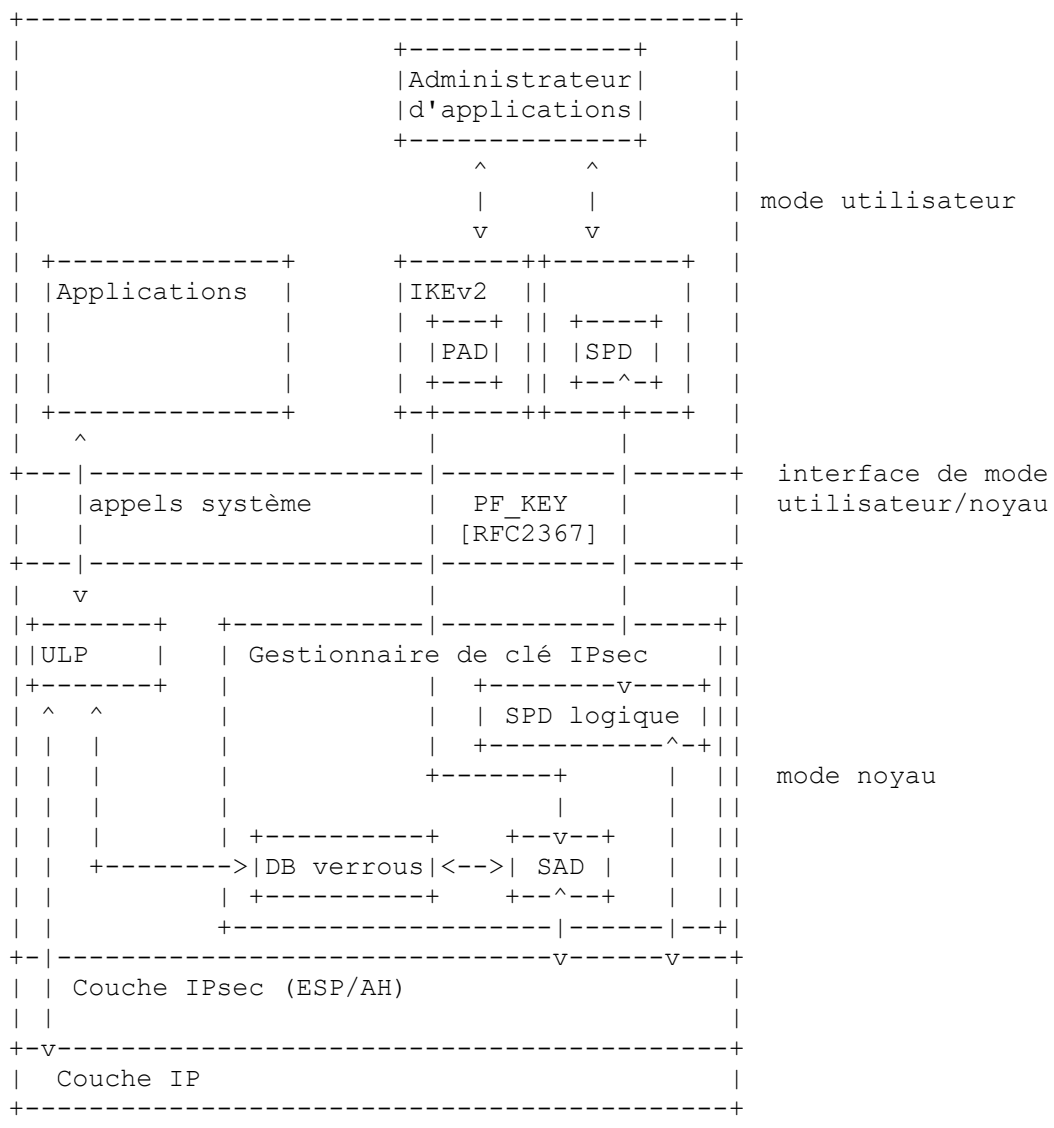

Figure 2 : Diagramme d'architecture de mise en œuvre pour information

Les interfaces d'ULP à la LD IPsec sont comme suit :

- CREATE_LISTENER_LATCH(triplet, [paramètres de type et qualité de protection]) -> bride de verrou | erreur
  Si il n'y a pas d'objet de verrouillage de connexion en conflit dans l'état LISTENER pour le triplet donné (adresse locale, protocole, et numéro d'accès local) et si la politique locale le permet, cette opération crée atomiquement un objet de verrouillage de connexion dans l'état LISTENER pour le triplet donné.
  Quand une SA fille est créée qui correspond au triplet d'un verrou écoutant, mais à aucun quintuplet (adresse locale, adresse distante, protocole, numéro d'accès local, et numéro d'accès distant) d'un verrouillage de connexion ESTABLISHED, le gestionnaire de clé crée alors un nouvel objet de verrouillage de connexion dans l'état ESTABLISHED. Le gestionnaire de clé DOIT informer le possesseur du verrou écoutant des verrouillages de connexion créés par suite du verrou d'écoutant ; voir l'interface "ALERT()" ci-dessous.

- CREATE_CONNECTION_LATCH(quintuplet, [paramètres de type et de qualité de protection], [identifiant d'homologue], [identifiant local]) -> bride de verrou | erreur
  Si a) le verrou demandé n'existe pas (ou existe, mais est dans l'état CLOSED) b) tous les paramètres de verrou sont fournis, ou si des SA convenables existent dans la SAD d'où elles sont déduites, et c) si il n'y a pas de conflit avec la SPD et la SAD, alors cela crée un verrouillage de connexion dans l'état ESTABLISHED. Si les paramètres de verrou ne sont pas fournis et si aucune SA convenable n'existe dans la SAD d'où déduire ces paramètres, alors le gestionnaire de clé DOIT initier des SA filles, et si besoin est, une SA IKE, d'où déduire ces paramètres.
  Le gestionnaire de clé PEUT retarder l'établissement de SA fille et retourner immédiatement après la vérification de politique, sachant que l'ULP qui a demandé le verrou produira ensuite un paquet qui va déclencher l'établissement de la SA. Une telle mise en œuvre peut exiger un état passager supplémentaire dans l'automate à états de verrouillage de connexion, un état "LARVAL", pour ainsi dire, qui n'est pas décrit ici.
  Si en fin de compte le verrouillage de connexion ne peut pas être établi, soit à cause de conflits, soit parce que aucune SA ne peut être établie avec l'homologue à l'adresse de destination, une erreur est alors retournée à l'ULP. (Si le gestionnaire de clé a retardé l'établissement de la SA, et si l'établissement de SA finit par échouer, le gestionnaire de clé doit alors informer l'ULP, éventuellement en asynchrone. Ceci est un des nombreux détails que les mises en œuvre qui utilisent un état LARVAL doivent prendre en compte.)

- RELEASE_LATCH(bride d'objet verrou)
  Change l'état du verrouillage de connexion donné en CLOSED ; le verrouillage de connexion est alors supprmé.
  Le gestionnaire de clé PEUT supprimer toutes les SA filles existantes qui correspondent au verrou donné si il a été dans l'état ESTABLISHED. Si le gestionnaire de clé supprime ces SA, il DEVRAIT alors informer l'homologue avec une charge utile d'information Delete (voir IKEv2 [RFC4306]).

- FIND_LATCH(quintuplet) -> bride de verrou | erreur
  Étant donné un quintuplet, retourne une bride de verrou (ou une erreur).

- INQUIRE_LATCH(bride d'objet verrou) -> {état de verrou, paramètres verrouillés} | erreur
  Retourne toutes les informations disponibles sur le verrou, incluant son état actuel (ou une erreur).

L'interface de LD IPsec à l'ULP est comme suit :

- ALERT(bride d'objet verrou, quintuplet, nouvel état, [raison])
  Alerte un ULP sur un changement d'état asynchrone pour le verrouillage de connexion donné et, facultativement, fournit la raison du changement.
  Cette interface est à fournir par chaque ULP au gestionnaire de clé. Les détails spécifiques de la façon dont cette interface est fournie sont spécifiques de la mise en œuvre, donc non spécifiés ici (par exemple, cela pourrait être une fonction de "rappel" ou de "clôture" enregistrée au titre de l'interface CREATE_LISTENER_LATCH() , ou elle pourrait être fournie quand l'ULP est chargé sur le système courant via une interface d'enregistrement fournie par le gestionnaire de clé).

  Inutile de dire que la LD est mise à jour chaque fois qu'est créé, supprimé, ou cassé un objet verrouillage de connexion.

  L'API décrite ci-dessus est un nouveau service du gestionnaire de clé IPsec. En particulier, le gestionnaire de clé IPsec DOIT empêcher les conflits entre verrouillages, et il DOIT empêcher les conflits entre tout verrou et des SA filles existantes ou proposées comme suit :

- Les verrouillages de connexion non écoutant NE DOIVENT PAS être créés si il existe des conflits de SA dans la SAD au moment de la demande de verrouillage de connexion ou de sa création (à partir d'un verrou écoutant). Une SA fille entre en conflit avec une autre, vue du verrou, si et seulement si : a) ses sélecteurs de trafic et les SA en conflit

correspondent au verrou donné, et b) ses paramètres d'homologue, de type de protection, ou de qualité de protection diffèrent de ceux de la SA en conflit.

- Les propositions de SA fille qui entreraient en conflit avec un verrouillage de connexion étendu et dont les sélecteurs de trafic peuvent être rétrécis pour éviter le conflit DEVRAIT être rétrécies (voir le paragraphe 2.9 de la [RFC4306]) ; autrement, le verrou DOIT transiter à l'état BROKEN.

- Lorsque des propositions de SA filles qui entreraient en conflit avec un verrouillage de connexion étendu ne peuvent pas être rétrécies pour éviter le conflit, le gestionnaire de clé DOIT casser le verrouillage de connexion et informer le possesseur (c'est-à-dire, l'ULP) avant d'accepter les SA en conflit.

Finalement, le gestionnaire de clé DOIT protéger les connections verrouillées contre des changements de SPD qui changeraient la qualité de protection accordée au trafic d'une connexion verrouillée, ou qui la contournerait. Quand un tel changement de configuration a lieu, le gestionnaire de clé DOIT répondre d'une des façons suivantes. Le comportement de mise en œuvre EXIGÉ est de passer à l'état BROKEN tous les verrouillages de connexion qui entrent en conflit avec le changement de SPD. Un comportement FACULTATIF est de mettre à jour logiquement la SPD comme si une entrée PROTECT avait été ajoutée à la tête du SPD-S avec des sélecteurs de trafic correspondant seulement au quintuplet de la connexion verrouillée, et avec les informations de traitement tirées du verrouillage de connexion. Ces mises à jour de la SPD NE DOIVENT PAS survivre au pannes ou réamorçage du système.

Les ULP créent des connexions verrouillées en s'interfaçant avec IPsec comme suit :

- Pour les points d'extrémité écoutants, l'ULP va demander un objet écoutant de verrouillage de connexion pour le triplet de l'écoutant ULP. Tous les paramètres de verrouillage demandés par l'application DOIVENT être passés.

- Quand l'ULP reçoit un paquet qui initie une connexion pour un quintuplet correspondant à un triplet de verrou d'écoutant, l'ULP va alors demander au gestionnaire de clé si un quintuplet de verrouillage de connexion a été créé. Sinon, l'ULP va soit rejeter la nouvelle connexion, soit l'accepter et informer l'application que la nouvelle connexion n'est pas verrouillée.

- Quand il initie une connexion, l'ULP va demander un objet de verrouillage de connexion pour le quintuplet de la connexion. Tous les paramètres de verrouillage demandés par l'application DOIVENT être passés. Si aucun verrouillage ne peut être créé, alors l'ULP DOIT soit retourner une erreur à l'application, soit continuer avec la nouvelle connexion et informer l'application que la nouvelle connexion n'est pas verrouillée.

- Quand une connexion est supprimée et qu'aucun autre paquet n'est attendu pour elle, l'ULP DOIT alors demander que l'objet de verrouillage de connexion soit détruit.

- Quand il supprime un écoutant, l'ULP DOIT demander que l'objet d'écoutant de verrouillage de connexion soit détruit.

- Quand un écoutant d'ULP rejette des connexions, l'ULP va demander la destruction de tous les objets de verrouillage de connexion qui ont pu être créés par suite de la tentative de l'homologue d'ouvrir la connexion.

- Quand le gestionnaire de clé informe un ULP qu'un verrouillage de connexion est passé à l'état BROKEN, l'ULP DOIT alors cesser d'envoyer des paquets et DOIT éliminer tous les paquets entrants suivants pour la connexion affectée jusqu'à ce qu'il revienne à l'état ESTABLISHED. Les ULP en mode connexion DEVRAIENT agir comme si la connexion subissait une perte de paquet.

- Quand le gestionnaire de clé informe un ULP qu'un verrouillage de connexion a été passé administrativement à l'état CLOSED, les ULP en mode connexion DOIVENT agir comme si la connexion avait été réinitialisée par l'homologue. Les mises en œuvre d'ULP qui ne sont pas en mode connexion, et qui n'ont pas d'API pour simuler une réinitialisation, DOIVENT éliminer tous les paquets entrants pour cette connexion et NE DOIVENT PAS envoyer d'autres paquets -- l'application est supposée détecter les fins de temporisations et agir en conséquence.

Le principal avantage de ce modèle de verrouillage de connexion est qu'il traite les mises en œuvre de IPsec lorsque le traitement de ESP/AH est mis en œuvre dans le matériel (pour toute la SAD de l'hôte ou un sous ensemble) même lorsque le matériel ne prend pas en charge l'étiquetage des paquets entrants avec les indices des entrées de SAD correspondant aux SA qui les protégeaient.

2.3.1 Conditions de concurrence et cas marginaux

Les ULP DOIVENT éliminer les paquets entrants et cesser d'envoyer des paquets immédiatement à réception d'un message de rupture de verrouillage de connexion. Autrement, l'ULP ne sera pas capable de distinguer les paquets entrants qui étaient protégés conformément au verrouillage de la connexion des paquets entrants qui ne l'étaient pas. Cela peut inclure d'éliminer des paquets entrants qui étaient protégés par une SA convenable ; éliminer de tels paquets n'est pas différent, du point de vue de l'ULP, d'une perte de paquet ailleurs sur le réseau à la couche IP ou en-dessous -- sans dommages, du point de vue de la sécurité, car la connexion est sécurisée, mais il peut en résulter des retransmissions.

Une autre condition de concurrence est comme suit. Un paquet TCP SYN protégé peut être reçu et désencapsulé, mais la SA qui le protégeait pourrait avoir expiré avant que le gestionnaire de clé ait créé le verrouillage de connexion qui aurait été créé par ce paquet. Dans ce cas, le gestionnaire de clé va devoir initier de nouvelles SA filles afin de déterminer l'identifiant d'homologue de l'envoyeur pour qu'il soit inclus dans le verrouillage de connexion. Ici, il n'est pas garanti que l'identifiant d'homologue pour les nouvelles SA soit le même que celui de l'homologue qui a envoyé le paquet TCP SYN. Cette condition de concurrence est sans dommages : TCP va envoyer un SYN+ACK au mauvais homologue, qui va alors répondre avec un RST *(réinitialiser)* -- le verrouillage de connexion va cependant refléter le nouvel homologue, de sorte que si le nouvel homologue est malveillant il ne sera pas capable d'apparaître comme étant l'ancien homologue. Donc, cette condition de concurrence est sans dommages.

2.3.2 Exemple

Considérons plusieurs systèmes avec une PAD très simple contenant une seule entrée comme :

Règle	SA fille Identifiant distant	Identifiants permis	Recherche de SPD
1	<tout valide pour l'ancre de confiance X>	192.0.2/24	par IP

Figure 3 : Exemple de PAD

Et une simple SPD comme :

Règle	TS local	TS distant	Prochain protocole	Action
1	192.0.2/24:ANY	192.0.2/24:1-5000	TCP	PROTECT(ESP,...)
1	192.0.2/24:1-5000	192.0.2/24:ANY	TCP	PROTECT(ESP,...)
1	ANY	ANY	ANY	BYPASS

Figure 4 : Tableau [SG-A] SPD

Cela dit effectivement : pour les accès TCP 1-5000 dans notre réseau, permettre seulement les homologues qui ont les accréditifs produits par la CA X et PROTÉGER ce trafic avec ESP, autrement, éviter tout autre trafic.

Considérons maintenant deux hôtes, A et B, dans ce réseau, qui souhaitent communiquer en utilisant l'accès 4000, et un troisième hôte, C, qui est aussi dans le même réseau et souhaite attaquer A et/ou B. Les trois hôtes ont les accréditifs et les certificats produits par la CA X. Imaginons aussi que A soit connecté à son réseau via une liaison sans fil et que son adresse soit allouée de façon dynamique.

B écoute sur l'accès 4000. A initie une connexion à partir de l'accès 32800 à B sur l'accès 4000.

On va supposer qu'il n'y a pas d'API IPsec, mais que TCP crée des verrouillages lorsque possible.

On va considérer trois cas : a) A et B prennent tous deux en charge le verrouillage de connexion, b) seul A le fait, c) seul B le fait. Pour les besoins de cet exemple, la SAD est vide sur les trois hôtes quand A initie sa connexion TCP à B sur l'accès 4000.

Quand une application fonctionnant sur A initie une connexion TCP à B sur l'accès 4000, A va commencer par créer un verrouillage de connexion. Comme la SAD est vide, A va initier un échange IKEv2 pour créer une SA IKE avec B et une paire de SA filles pour le quintuplet {TCP, A, 32800, B, 4000}, puis un nouveau verrouillage va être créé dans l'état ESTABLISHED. Un peu plus tard, TCP va envoyer un paquet SYN protégé par la SA fille A-à-B, d'après la SPD.

Quand une application fonctionnant sur B crée une "prise" d'écoutant TCP sur l'accès 4000, B va créer un verrouillage de connexion LISTENER pour le triplet {TCP, B, 4000}. Quand B reçoit le paquet SYN TCP de A, il va alors créer un

verrouillage de connexion pour {TCP, B, 4000, A, 32800}. Comme à ce moment, des SA filles ont été créées dont les sélecteurs de trafic englobent le quintuplet et qu'il n'y a pas d'autres SA en conflit dans la SAD, ce verrouillage de connexion va être créé dans l'état ESTABLISHED.

Si C tente de monter une attaque par interposition sur A (c'est-à-dire, prétend avoir la ou les adresses de B) après que A a créé son verrouillage de connexion, alors les SA de C avec A vont causer la rupture du verrouillage de connexion, et la réinitialisation de la connexion TCP (parce qu'on suppose qu'il n'y a pas d'API par laquelle TCP pourrait notifier à l'application la rupture du verrouillage de connexion). Si C tente de se faire passer pour A auprès de B, alors la même chose va arriver sur B.

Si A ne prend pas en charge le verrouillage de connexion, alors C va être capable de se faire passer pour B auprès de A à tout moment. Sans avoir vu le trafic non chiffré entre A et B, C être limité par les numéros de séquence de TCP à des attaques de style RST. De même, si B ne prend pas en charge le verrouillage de connexion, alors C va être capable de se faire passer pour A auprès de B.

2.4 Modèle pour information : étiquetage de paquet local

Dans ce paragraphe, on décrit le verrouillage de connexion en termes d'interfaces entre les ULP et IPsec sur la base de l'étiquetage des paquets lorsque ils montent et descendent dans la pile IP. Ce paragraphe est pour INFORMATION.

Dans ce modèle, les ULP maintiennent des objets et des états de verrouillage de connexion, plutôt que le gestionnaire de clé IPsec, et mettent en antémémoire un sous ensemble de SPD décorrélé dans les TCB d'ULP. L'étiquetage des paquets, quand ils montent et descendent dans la pile, avec les identifiants de SA, permet alors aux ULP d'appliquer la sémantique de verrouillage de connexion. Bien sûr, ces étiquette n'apparaissent pas dans le réseau.

L'interface entre les ULP et l'interface IPsec est comme suit :

- La couche IPsec étiquette tous les paquets protégés entrants adressés à l'hôte avec l'indice de l'entrée de SAD correspondant à la SA qui protégeait le paquet.

- La couche IPsec comprend deux types d'étiquettes sur les paquets sortants :
  * une étiquette qui spécifie un ensemble de paramètres verrouillés (identifiant d'homologue, qualité de protection, etc.) que la couche IPsec va utiliser pour trouver ou acquérir une SA appropriée pour protéger le paquet sortant (autrement IPsec va informer l'ULP et éliminer le paquet) ;
  * une étiquette demandant une rétroaction sur la SA utilisée pour protéger le paquet sortant, si il en est.

Les ULP créent des connexions verrouillées en s'interfaçant comme suit avec IPsec :

- Quand l'ULP passe un paquet initiateur d'une connexion à IP, l'ULP demande des retours sur la SA utilisée pour protéger le paquet sortant, si il en est, et peut spécifier les paramètres de verrouillage demandés par l'application. Si le paquet est protégé par IPsec, alors l'ULP enregistre certains paramètres de la SA utilisée pour le protéger dans le TCB de la connexion.

- Quand un ULP reçoit un paquet initiateur d'une connexion, il traite l'étiquette IPsec du paquet, et il enregistre dans le TCB de la connexion les paramètres de la SA qui devraient être verrouillés.

Une fois que les paramètres de SA sont enregistrés dans le TCB d'une connexion, l'ULP applique le verrou de la connexion, ou les liens, à ces paramètres comme suit :

- L'ULP traite l'étiquette IPsec de tous les paquets entrants pour une connexion donnée et vérifie que les SA utilisées pour protéger les paquets entrants correspondent aux verrouillages de connexion enregistrés dans les TCB. Les paquets qui ne sont pas protégés ainsi sont éliminés (cela correspond à faire transiter le verrouillage de connexion à l'état BROKEN jusqu'à ce que le prochain paquet acceptable arrive, mais dans ce modèle, cette transition est imaginaire) ou causent la rupture par l'ULP du verrouillage de connexion et l'information de l'application.

- L'ULP demande toujours que les paquets sortants soient protégés par des SA qui correspondent à la connexion verrouillée en étiquetant de façon appropriée les paquets sortants.

En mettant effectivement en antémémoire un sous ensemble de la SPD décorrélée dans les TCB de l'ULP et par cette nature d'étiquetage de paquet, cette méthode de verrouillage de connexion peut aussi optimiser le traitement de la SPD en

suppléant au besoin de chercher, à la fois en entrée et en sortie, les paquets destinés à l'hôte ou générés par l'hôte. Cela fait de la mise en œuvre de la mise à jour FACULTATIVE de "SPD logique" décrite aux paragraphes 2.3 et 3.1 un effet collatéral incident de cette approche.

Ce modèle de verrouillage de connexion peut n'être pas utilisable avec les matériels ESP/AH qui ne prennent pas en charge le schéma d'étiquetage de paquet décrit ci-dessus.

Noter que ce modèle n'a pas d'état explicite BROKEN de verrouillage de connexion.

Étendre l'interface d'étiquetage de paquet ULP/IPsec à l'application pour l'utiliser avec des transports de datagrammes sans connexion devrait permettre aux applications d'utiliser de tels transports et de mettre en œuvre le verrouillage de connexion à la couche application.

2.5 IPsec en mode non natif

Ce paragraphe est pour INFORMATION.

Les mises en œuvre non natives d'IPsec, principalement BITS et SG, peuvent aussi mettre en œuvre le verrouillage de connexion. Une distinction majeure entre IPsec natif et BITS, pris sur le réseau (BITW, *bump-in-the-wire*) ou IPsec SG est l'absence d'API pour les applications aux points d'extrémité dans le cas du dernier. Par suite, il peut n'y avoir aucune utilisation des interfaces de gestion de verrou décrites au paragraphe 2.3 : pas aux points d'extrémité de l'ULP. Donc, les mises en œuvre de BITS/BITW/SG doivent discerner l'état de connexion d'ULP de l'inspection de paquet (que de nombreux pare-feu peuvent faire) et émuler les appels au gestionnaire de clé en conséquence.

Quand un verrouillage de connexion est rompu, une mise en œuvre de BITS/BITW/SG peut avoir à simuler une réinitialisation de connexion en envoyant les paquets appropriés (par exemple, paquets des TCP RST) pour les connexions affectées.

Comme avec tous les boîtiers de médiation à états pleins, ce schéma souffre de l'incapacité du boîtier de médiation à interagir avec les applications. Par exemple, la mort de la connexion peut être difficile à rendre certaine. Les applications de lien de canal ne peuvent pas non plus fonctionner avec les canaux maintenus par des mandataires sans être capables de communiquer (de façon sûre) sur lui avec le boîtier de médiation.

2.6 Note de mise en œuvre sur les identifiants d'homologues

Une des recommandations pour les mises en œuvre de verrouillage de connexion est de rendre les charges utiles CERT (certificats) d'homologue disponibles aux applications.

De plus, les clés publiques brutes sont probablement à utiliser dans la construction de liens de canal pour les canaux IPsec (voir [IPSEC-CB]) et elles doivent être disponibles, dans tous les cas, afin de mettre en œuvre un "acte de foi" à la couche d'application (voir les [RFC5386] et [RFC5387]).

Les certificats et les clés publiques brutes sont de grandes chaînes binaires, trop grandes pour être raisonnablement conservées dans les mises en œuvre en mode noyau de verrouillage de connexion (qui vont probablement être le cas normal). De telles mises en œuvre devraient conserver les identifiants d'homologues dans une base de données en mode utilisateur et utiliser de petits entiers pour se référer à eux à partir de la SAD et LD en mode noyau. La corruption d'une telle base de données ressemble à la corruption de la SAD/LD ; en cas de corruption, la mise en œuvre DOIT agir comme si tous les verrouillages de connexion ESTABLISHED et BROKEN étaient passés administrativement à l'état CLOSED. Les mises en œuvre sans API IPsec PEUVENT hacher les identifiants d'homologue et utiliser le hachage pour se référer à eux, de préférence en utilisant un algorithme de hachage fort.

3. Caractéristiques facultatives

Au strict minimum, le verrouillage de connexion est une couche passive par dessus IPsec, qui avertit les ULP de changements de SPD et SAD incompatibles avec l'état de SPD/SAD qui était applicable à une connexion quand il a été établi.

Il y a des caractéristiques facultatives, comme des API (abstraites). Certaines de ces caractéristiques rendent le verrouillage

de connexion une caractéristique un peu plus active. Spécifiquement, les mises à jour de SPD logique facultatives décrites au paragraphe 2.3 et caractéristique de protection/outrepassement facultatif décrite au paragraphe suivant.

3.1 Protection facultative

Avec des API IPsec, une application pourrait demander qu'un paquet d'une connexion soit protégé alors qu'il serait autrement outrepassé ; c'est-à-dire, les applications pourrait outrepasser la politique BYPASS. Les applications privilégiées localement pourraient demander que les paquets de leurs connexions soient outrepassés plutôt que protégés ; c'est-à-dire, des applications privilégiées pourraient outrepasser la politique PROTECT. On appelle cela "protection facultative".

Les deux modèle de verrouillage de connexion de IPsec natif peuvent être étendus pour prendre en charge la protection facultative. Avec le modèle décrit au paragraphe 2.4, la protection facultative vient naturellement : la couche IPsec a seulement besoin de vérifier que la protection demandée pour les paquets sortants satisfait ou excède (comme déterminé par la politique locale ou du système) la qualité de protection, si il en est, exigée par la SPD. Dans le cas du modèle décrit au paragraphe 2.3, l'application des exigences de protection minimum serait faite par le gestionnaire de clé IPsec via l'automate à états de verrouillage de connexion.

Quand une application demande, et que la politique locale le permet, une protection supplémentaire ou l'outrepassement de protection, la SPD DOIT être logiquement mise à jour afin qu'il existe une entrée de SPD convenable pour protéger ou outrepasser le quintuplet exact enregistré par le verrouillage de connexion correspondant. Ces mises à jour logiques de SPD DOIVENT être faites au moment de la création du verrouillage de connexion, et DOIVENT être faites atomiquement (voir la note sur les conditions de concurrence au paragraphe 2.3). Ces mises à jour de la SPD NE DOIVENT PAS survivre aux pannes ou réamorçages du système.

4. Établissement simultané de verrouillage

Certains ULP en mode connexion, spécifiquement TCP, prennent en charge simultanément des connexions (où deux clients se connectent à chaque autre, en utilisant le même quintuplet, en même temps). Le verrouillage de connexion prend en charge aussi le verrouillage simultané, pourvu que le protocole d'échange de clé ne le rende pas impossible.

Si on considère deux applications faisant une connexion TCP simultanée l'une à l'autre et qui demandent un canal IPsec. Si elles demandent les mêmes paramètres de verrouillage de connexion, alors la connexion et le canal devraient être établis comme d'habitude. Même si le protocole d'échange de clés utilisé ne prend pas en charge l'établissement simultané de IKE_SA et/ou de SA filles, pourvu qu'une tentative de l'homologue de créer les SA filles nécessaires réussisse, alors l'autre homologue devrait être capable de remarquer les nouvelles SA immédiatement lorsque il échoue dans sa tentative de les créer lui-même.

Si, cependant, les deux applications homologues devaient demander des paramètres différents de verrouillage de connexion, alors le verrouillage de connexion doit échouer sur une des extrémités ou sur les deux.

5. Verrouillage de connexion à IPsec pour divers ULP

Les paragraphes qui suivent décrivent le verrouillage de connexion pour chacun des trois protocoles de transport. Noter que pour TCP et UDP, il n'y a rien dans les paragraphes qui suivent qui ne devrait pas déjà être évident à partir du reste du présent document. Le paragraphe sur SCTP, cependant, spécifie des détails relatifs au multi-rattachements SCTP, qui peuvent n'être pas aussi évidents.

5.1 Verrouillage de connexion à IPsec pour TCP

La création/suppression de verrouillage de connexion IPsec pour les connexions TCP [RFC0793] est déclenchée quand :

- un point d'extrémité d'écoutant TCP est créé (par exemple, quand la fonction BSD Sockets listen() est invoquée sur une prise). Cela devrait causer la création d'un verrouillage de connexion LISTENER.
- un paquet TCP SYN est reçu sur une adresse IP et un numéro d'accès pour lequel il y a un écoutant. Cela devrait causer la création d'un verrouillage de connexion ESTABLISHED ou BROKEN.
- un paquet TCP SYN est envoyé (par exemple, par suite d'une invocation de la fonction BSD Sockets connect()). Cela devrait causer la création d'un verrouillage de connexion ESTABLISHED ou BROKEN.

- toute transition d'état d'une connexion TCP à l'état CLOSED va causer aussi une transition correspondante pour tout verrouillage de connexion associé à l'état CLOSED.

Voir le paragraphe 5.5 sur comment traiter les transitions de verrouillage à l'état BROKEN.

5.2 Verrouillage de connexion à IPsec pour UDP avec des connexions simulées

UDP [RFC0768] est un protocole de transport sans connexion. Cependant, certaines API de réseautage (par exemple, l'API de prise de BSD) permettent l'émulation de connexions UDP. Dans ce cas, le verrouillage de connexion peut être pris en charge en utilisant l'un ou l'autre des modèles donnés ci-dessus. On ignore, dans ce paragraphe, le fait que le modèle de verrouillage de connexion décrit au paragraphe 2.4 puisse prendre en charge le verrouillage par datagramme en étendant ses interfaces d'étiquetage de paquets à l'application.

La création/suppression de verrouillage de connexion IPsec pour les connexions UDP est déclenchée quand :
- une application crée une "connexion" UDP. Cela devrait causer la création d'un verrouillage de connexion ESTABLISHED ou BROKEN.
- une application détruit une "connexion" UDP. Cela devrait causer la création d'un verrouillage de connexion ESTABLISHED ou BROKEN.

Quand un verrouillage de connexion transite à l'état BROKEN et que l'application demandait (ou que la politique du système impose) que la connexion soit rompue, alors UDP devrait informer l'application, si il y a un moyen de le faire, ou autrement il devrait attendre, permettant à la stratégie de durée de vie/temporisation de la couche application, si il y en a une, d'amener la connexion en fin de temporisation.

Ce qui constitue une action appropriée en présence de transitions administratives de verrouillages de connexion à l'état CLOSED dépend de si l'API de prise UDP "connected" de la mise en œuvre fournit un moyen pour que la prise retourne une erreur indiquant qu'elle a été close.

5.3 Verrouillage de connexion à IPsec pour UDP avec API d'étiquetage de datagrammes

Les mises en œuvre fondées sur l'un ou l'autre modèle de verrouillage de connexion peuvent fournir aux applications des API d'étiquetage de datagramme fondées sur celles décrites au paragraphe 2.4. Les mises en œuvre UDP avec le modèle normatif de verrouillage de connexion IPsec doivent confirmer, en sortie, que le quintuplet fourni par l'application est en accord avec le verrouillage de connexion fourni par l'application ; en entrée, UDP peut déduire l'étiquette en cherchant un verrouillage de connexion correspondant au quintuplet du datagramme entrant.

5.4 Verrouillage de connexion à IPsec pour SCTP

SCTP [RFC4960], un protocole en mode connexion est, d'une certaine façon, similaire à TCP. La plus grande différence, à l'égard du verrouillage de connexion, entre SCTP et TCP est que SCTP permet à chaque point d'extrémité d'être identifié par un ensemble d'adresses IP, bien que, comme dans TCP, chaque point d'extrémité d'une connexion SCTP (ou, plutôt, d'une association SCTP) puisse seulement avoir un numéro d'accès.

On peut représenter la pluralité des adresses de point d'extrémité d'association SCTP comme une pluralité de quintuplets, dont chacun a son propre verrouillage de connexion. Autrement, on peut étendre l'objet de verrouillage de connexion pour prendre en charge plusieurs adresses pour chaque point d'extrémité. La première approche est utilisée dans le présent document ; donc, on va supposer cette représentation.

Bien sûr, cette approche résulte en N x M verrouillages de connexion pour toute association SCTP (où un point d'extrémité a N adresses et l'autre en a M) ; tandis que la solution de remplacement exige un verrouillage de connexion par association SCTP (avec N + M adresses). Les mises en œuvre peuvent choisir l'une ou l'autre approche.

La création/suppression de verrouillage de connexion IPsec pour les connexions SCTP est déclenchée quand :
- un point d'extrémité écoutant SCTP est créé (par exemple, quand la fonction de prise SCTP listen() est invoquée sur une prise). Cela devrait causer la création d'un verrouillage de connexion LISTENER pour chaque adresse de l'écoutant.
- un tronçon SCTP INIT est reçu sur une adresse IP et numéro d'accès pour lequel il y a un écoutant. Cela devrait causer la création d'un ou plusieurs verrouillages de connexion ESTABLISHED ou BROKEN, un pour chaque quintuplet distinct par adresse de client et serveur.
- un tronçon SCTP INIT est envoyé (par exemple, par suite d'une invocation de la fonction de prises SCTP connect()).

Cela devrait causer la création d'un ou plusieurs verrouillages de connexion ESTABLISHED ou BROKEN.

- un tronçon SCTP Changement de configuration d'adresse (ASCONF, *Address Configuration Change Chunk*) [RFC5061] ajoutant une adresse IP de point d'extrémité est envoyé ou reçu. Cela devrait causer la création d'un ou plusieurs verrouillages de connexion ESTABLISHED ou BROKEN.
- toute transition d'état d'une association SCTP à l'état CLOSED va causer aussi une transition correspondante pour tout verrouillage de connexion associé à l'état CLOSED.
- un tronçon SCTP ASCONF [RFC5061] supprimant l'adresse IP d'un point d'extrémité est envoyé ou reçu. Cela devrait causer la clôture d'un ou plusieurs verrouillages de connexion associés.

Voir au paragraphe 5.5 comment traiter les transitions de verrouillage à l'état BROKEN.

5.5 Traitement de l'état BROKEN pour TCP et SCTP

Il y a plusieurs façons de traiter les transitions de verrouillage de connexion à l'état BROKEN dans le cas d'ULP en mode connexion comme TCP ou SCTP :

a. Attendre un possible retour futur à l'état ESTABLISHED, l'ULP ne déplaçant aucune donnée entre les deux points d'extrémité de la connexion pendant ce temps. Les mécanismes de temporisation d'ULP et d'application vont, bien sûr, être déclenchés au cas où la station à l'état BROKEN serait trop longue. SCTP peut détecter ces fins de temporisation et initier une reprise sur défaillance, dans le cas d'associations multi-rattachements.

b. Agir comme si la connexion avait été réinitialisée (un message RST est reçu, dans TCP, ou un message ABORT, dans SCTP).

c. Agir comme si un message ICMP Destination injoignable avait été reçu (dans SCTP ces messages peuvent déclencher la reprise de chemin sur défaillance dans le cas d'associations multi-rattachements).

Les mises en œuvre DEVRAIENT fournir des API qui permettent aux applications soit 1) d'être informées (en asynchrone ou autrement) de rupture de verrouillage afin qu'elles puissent choisir une disposition, et/ou 2) de choisir une disposition spécifique a priori (avant que survienne une rupture de verrouillage). Les options de disposition sont attendre, clore, ou poursuivre avec la reprise de chemin sur défaillance.

Les mises en œuvre DOIVENT fournir une disposition par défaut en cas de rupture de verrouillage de connexion. Bien que (a) soit clairement le comportement par défaut le plus pur, on RECOMMANDE (b) pour les associations TCP et SCTP lorsque il reste seulement un chemin (un quintuplet), et (c) pour les associations SCTP multi-rattachements. La raison de cette recommandation est comme suit : une SA en conflit indique très probablement que l'homologue d'origine est parti et a été remplacé par un autre, et il est peu probable que l'homologue original revienne ; donc, reprendre plus vite sur défaillance semble raisonnable.

Noter que notre comportement par défaut recommandé ne crée pas d'attaques de déni de service par réinitialisation hors chemin . Pour casser un verrouillage de connexion, un attaquant devrait d'abord avoir réussi à établir une SA, avec un des points d'extrémité de la connexion, qui est en conflit avec le verrouillage de connexion et qui exige l'échange de plusieurs messages entre ce point d'extrémité et l'attaquant. Sauf si le point d'extrémité choisi comme victime par l'attaquant lui permet de revendiquer la gamme d'adresses IP pour ses SA, l'attaquant devrait s'emparer des adresses des autres points d'extrémité, ce qui exclut les attaques hors chemin

6. Considérations sur la sécurité

6.1 Impact sur IPsec

Le verrouillage de connexion ajoute effectivement un mécanisme pour traiter l'existence, dans la SAD, de plusieurs SA filles non équivalentes avec des sélecteurs de trafic en chevauchement. Ce mécanisme consiste en, au minimum, une notification locale des protocoles de transport (et, par eux, des applications) de l'existence d'un conflit qui affecte les connexions d'une couche de transport. Les transports affectés sont aussi notifiés quand le conflit est réglé. Les transports doivent éliminer les paquets entrants, et ne doivent pas envoyer de paquets sortants pour les connexions qui sont affectées par un conflit. Dans cette forme minimale, le verrouillage de connexion est une caractéristique passive, locale, mise en couche par dessus IPsec.

On réalise cela en ajoutant un nouveau type de base de données IPsec, la base de données de verrouillage (LD, *Latch*

Database) contenant les objets qui représentent l'intérêt d'un protocole de transport à protéger un flux de paquets donné de tels conflits. La LD est gérée en conjonction avec les mises à jour de la SAD et de la SPD, de sorte que les mises à jour à l'une ou l'autre qui entrent en conflit avec des verrouillages de connexion établis puissent être détectés. Pour certaines mises en œuvre de IPsec, cela peut impliquer des changements significatifs à leur constitution. Cependant, deux différents modèles de verrouillage de connexion sont donnés, et on espère que la plupart des mises en œuvre natives de IPsec vont trouver au moins un modèle assez simple pour être mis en œuvre dans leur pile de protocoles.

Cette notion de conflit de SA et de comment traiter la situation ne modifie pas l'architecture IPsec de base -- les caractéristiques de IPsec qui permettent que de tels conflits apparaissent demeurent, et il appartient aux protocoles de transport et aux applications de choisir si et comment leur répondre.

Il y a cependant des cas particuliers intéressants dans le modèle normatif de verrouillage de connexion dont les développeurs doivent avoir connaissance. Les notes du paragraphe 2.3.1 sont particulièrement pertinentes.

6.2 Impact sur IPsec des caractéristiques facultatives

La Section 3 décrit des caractéristiques facultatives de verrouillage de connexion où le gestionnaire de clé joue un rôle un peu plus actif, bien que toujours local. Il y a deux de ces caractéristiques : la protection/outrepassement facultative et la préservation des entrées "logiques" de SPD pour permettre que les connexions verrouillées restent dans l'état ESTABLISHED en présence des changements administratifs contraires de SPD (mais pas de SAD). Ces deux caractéristiques interagissent avec les interfaces administratives à IPsec ; les administrateurs doit avoir connaissance de ces caractéristiques, et il DEVRAIT y avoir un moyen de rompre les verrouillages de connexion dans l'état ESTABLISHED. Aussi, compte tenu des tendances récentes vers des parties centralisées de politique IPsec, ces deux caractéristiques peuvent être dites avoir des effets non locaux qui empêchent les changements distribués de politique de s'appliquer complètement.

6.3 Considérations sur la sécurité pour les applications

Le verrouillage de connexion n'est pas négocié. Il est donc possible à une extrémité d'une connexion d'utiliser le verrouillage de connexion alors que l'autre ne l'utilise pas ; dans ce cas, il est possible à des changements de politique locaux à l'extrémité non verrouillée de causer l'envoi de paquets non protégés. L'extrémité qui fait le verrouillage de connexion va rejeter les paquets non protégés, mais si ils portent des données sensibles, le dommage peut alors être déjà causé. Donc, les applications DEVRAIENT vérifier que les deux extrémités d'une connexion sont verrouillées (une telle vérification est implicite pour les applications qui utilisent le lien de canal à IPsec).

Le verrouillage de connexion protège les connexions individuelles du lien faible entre identifiant d'homologue et adresse, des changements de configuration IPsec, et des configurations qui permettent que plusieurs homologues affirment la même adresse. Mais le verrouillage de connexion n'assure pas qu'une des deux connexions avec la même adresse de point d'extrémité va avoir le même identifiant d'homologue verrouillé. En d'autres termes, les applications qui utilisent plusieurs connexions concurrentes entre deux nœuds donnés peuvent n'être plus protégées ou l'être moins en utilisant le verrouillage de connexion IPsec que par l'utilisation de IPsec seul sans verrouillage de connexion. De telles applications multi-connexions peuvent, cependant, examiner les paramètres de SA verrouillée de chaque connexion pour s'assurer que toutes les connexions concurrentes avec la même adresse de point d'extrémité ont aussi le même identifiant de point d'extrémité IPsec.

Le verrouillage de connexion protège contre les attaques de RST TCP. Cela n'aide cependant pas, si l'homologue original d'une connexion TCP n'est plus disponible (par exemple, si un attaquant a été capable d'interrompre la connexion réseau entre les deux homologues).

6.4 Lien de canal et API IPsec

Les canaux IPsec sont un prérequis du lien de canal [RFC5056] à IPsec. Le verrouillage de connexion fournit de tels canaux, mais les liens de canal pour les canaux IPsec (connexions verrouillées) ne sont pas spécifiés ici -- c'est un travail en cours [IPSEC-CB].

Sans API IPsec, le verrouillage de connexion fournit des avantages de sécurité marginaux par rapport à l'IPsec traditionnel. De telles API ne sont pas décrites ici ; voir [ABSTRACT-API].

6.5 Attaques de déni de service

Les transitions d'état de verrouillage de connexion à l'état BROKEN peuvent être déclenchées par des attaquants dans le chemin et tous les attaquants hors chemin qui peuvent attaquer les routeurs ou faire qu'un point d'extrémité accepte un message Redirect ICMP. Le verrouillage de connexion protège les applications contre des attaquants en chemin et hors chemin en général, mais pas spécifiquement contre le déni de service en chemin.

Des attaquants peuvent rompre les verrouillages si ils peuvent déclencher la DPD sur un point d'extrémité ou les deux et si ils causent l'immobilité des paquets entre deux points d'extrémité. De telles attaques exigent généralement que l'attaquant soir en chemin ; donc, on considère qu'il est acceptable de rompre les verrouillages quand DPD conclut qu'un homologue est mort ou réamorcé.

Des attaquants peuvent aussi rompre les verrouillages si la politique IPsec sur un nœud permet aux attaquant d'utiliser l'adresse IP d'un homologue de ce nœud. De telles configurations sont supposées être utilisées en conjonction avec BTNS en général. Ces attaques exigent généralement que l'attaquant soit en chemin.

7. Remerciements

L'auteur remercie Michael Richardson de son aide, ainsi que Stephen Kent, Sam Hartman, Bill Sommerfeld, Dan McDonald, Daniel Migault, et de nombreux autres qui ont participé au groupe de travail BTNS ou qui ont répondu aux questions sur IPsec, les mises en œuvre de verrouillage de connexion, etc.

8. Références

8.1 Références normatives

[RFC0768] J. Postel, "Protocole de datagramme d'utilisateur (UDP)", (STD 6), 28 août 1980.

[RFC0793] J. Postel (éd.), "Protocole de commande de transmission – Spécification du protocole du programme Internet DARPA", STD 7, DOI 10.17487/RFC0793, septembre 1981. (*Remplacée par* RFC9293)

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

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

[RFC4306] C. Kaufman, "Protocole d'échange de clés sur Internet (IKEv2)", décembre 2005. *(Obsolète, voir la* RFC5996, *remplacée par la* RFC7296)

[RFC4960] R. Stewart, éd., "Protocole de transmission de commandes de flux (SCTP)", septembre 2007. (*Remplace* RFC2960, RFC3309 ; *P.S. ; Remplacée par* RFC9260)

[RFC5061] R. Stewart et autres, "Reconfiguration dynamique d'adresse pour le protocole de transmission de contrôle de flux (SCTP)", septembre 2007. *(P.S.)*

[RFC5386] N. Williams, M. Richardson, "La sécurité mieux que rien : un mode non authentifié de IPsec", novembre 2008. *(P.S.)*

8.2 Références pour information

[ABSTRACT-API] Richardson, M., "An abstract interface between applications and IPsec", Travail en cours, novembre 2008.

[IPSEC-CB] Williams, N., "End-Point Channel Bindings for IPsec Using IKEv2 and Public Keys", Travail en cours, avril 2008.

[IP_SEC_OPT] Sun Microsystems, Inc., "ipsec(7P) man page, Solaris 10 Reference Manual Collection".

[RFC1034] P. Mockapetris, "Noms de domaines - Concepts et facilités", STD 13, novembre 1987. (*MàJ par* RFC1101, 1183, 1348, 1876, 1982, 2065, 2181, 2308, 2535, 4033, 4034, 4035, 4343, 4035, 4592, 5936, 8020, 8482, 8767)

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

[RFC5056] N. Williams, "Sur l'utilisation des liens de canaux pour sécuriser les canaux", novembre 2007. *(P.S.)*

[RFC5387] J. Touch et autres, "Problème et déclaration d'applicabilité pour la sécurité mieux que rien (BTNS)", novembre 2008. *(Information)*

[RFC5406] S. Bellovin, "Lignes directrices pour spécifier l'utilisation de IPsec version 2", février 2009. (BCP0146)

[USING-IPSEC] Dondeti, L. and V. Narayanan, "Guidelines for using IPsec and IKEv2", Travail en cours, octobre 2006.

Adresse de l'auteur

Nicolas Williams
Sun Microsystems
5300 Riata Trace Ct
Austin, TX 78727
US
mél : Nicolas.Williams@sun.com