Groupe de travail Réseau
Request for Comments : 5082
RFC rendue obsolète : 3682
Catégorie : Sur la voie de la normalisation

Traduction Claude Brière de L'Isle

V. Gill
J. Heasley
D. Meyer
P. Savola, éditeur
C. Pignataro
octobre 2007

Mécanisme de sécurité TTL généralisé (GTSM)

Statut du présent mémoire

Résumé

L'utilisation du champ Durée ce vie (TTL, *Time to Live*) (IPv4) ou Limite de bonds (IPv6) d'un paquet, pour vérifier si le paquet a été généré par un nœud adjacent sur une liaison connectée, a été mise en œuvre dans de nombreux protocoles récents. Le présent document généralise cette technique. Le présent document rend obsolète la RFC 3682 expérimentale.

Table des Matières

1. Introduction

Le mécanisme de sécurité TTL généralisé (GSTM, *Generalized TTL Security Mechanism*) est conçu pour protéger le plan de contrôle fondé sur IP d'un routeur contre les attaques fondées sur l'utilisation de la CPU. En particulier, alors que les techniques cryptographiques peuvent protéger l'infrastructure fondée sur le routeur (par exemple, BGP [RFC4271], [RFC4272]) contre des attaques très variées, de nombreuses attaques fondées sur la surcharge de CPU peuvent être empêchées par le simple mécanisme décrit dans le présent document. Noter que la même technique protège contre d'autres attaques par raréfaction de ressource impliquant la CPU d'un routeur, telles que des attaques contre la bande passante de carte de ligne de traitement.

GTSM se fonde sur le fait que la vaste majorité des échanges de trafic de protocole sont établis entre des routeurs adjacents. Donc, la plupart des échanges de trafic de protocole sont soit directement entre des interfaces connectées, soit, dans le pire des cas, entre deux boucles de retour, avec des chemins statiques pour les boucles de retour. Comme l'usurpation de TTL est considérée comme presque impossible, un mécanisme fondé sur une valeur de TTL attendue peut fournir une défense

simple et raisonnablement robuste contre des attaques de l'infrastructure fondées sur des paquets de protocole falsifiés provenant de l'exérieur du réseau. Noter cependant, que GTSM n'est pas un substitut des mécanismes d'authentification. En particulier, il ne sécurise pas contre des attaques d'interposition dans le réseau, comme des paquets contrefaits ou répétés.

Finalement, le mécanisme GTSM est également applicable au TTL (IPv4) et à la limite de bonds (IPv6), et du point de vue de GTSM, TTL et limite de bonds ont une signification identique. Par suite, dans le reste de ce document le terme "TTL" est utilisé pour se référer aussi bien au TTL qu'à la limite de bonds (comme approprié).

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

2. Hypothèses sous jacentes à GTSM

GTSM est construit sur les hypothèses suivantes :

1. La vaste majorité des échanges de trafic de protocole est entre des routeurs adjacents.

2. Les fournisseurs de services peuvent ou non configurer un filtrage d'entrée strict [RFC3704] sur les liaisons qui ne sont pas de confiance. Si une protection maximale est désirée, un tel filtrage est nécessaire comme décrit au paragraphe 2.2.

3. L'utilisation de GTSM est FACULTATIVE, et peut être configurée homologue par homologue (ou groupe d'homologues).

4. Les routeurs homologues mettent tous deux en œuvre GTSM.

5. Le routeur prend en charge une méthode pour utiliser des réservoirs de ressources séparés (par exemple, des files d'attente, des quotas de traitement) pour les différentes classes de trafic.

Noter que le présent document ne prescrit pas d'autres restrictions qu'un routeur pourrait appliquer aux paquets qui ne respectent pas les règles de filtrage de GTSM, comme d'éliminer les paquets qui ne corrrespondent pas à une session de protocole configurée et de limiter en débit le reste. Le présent document ne suggère pas non plus les moyens réels de séparation de ressources, car ils sont spécifiques du matériel et de la mise en œuvre.

Cependant, la possibilité de prévention d'attaque de déni de service (DoS) se fonde sur l'hypothèse que la classification des paquets et la séparation de leurs chemins sont faites avant que les paquets passent par une ressource rare dans le système. En pratique, plus le traitement GTSM est fait au matériel le plus proche du débit de ligne, plus le système est résistant aux attaques de DoS.

2.1 Négociation GTSM

Le présent document suppose que, quand il est utilisé avec des protocoles existants, GTSM va être configuré manuellement entre des protocoles homologues. C'est-à-dire, aucune négociation automatique de capacité GTSM, comme en est fournie par la [RFC3392], n'est supposée ni définie.

Si un nnouveau protocole est conçu avec la prise en charge incorporée de GTSM, il est alors recommandé que les procédures soient toujours utilisées pour envoyer et valider les paquets de protocole reçus (GTSM est toujours activé, voir par exemple la [RFC2461]). Si, cependant, une négociation dynamique de la prise en charge de GTSM est nécessaire, les messages de protocole utilisés pour une telle négociation DOIVENT être authentifiés en utilisant d'autres mécanismes de sécurité pour empêcher des attaques de DoS.

Noter aussi que la présente spécification n'offre pas de mécanisme générique de négociation de capacité GTSM, de sorte que les messages du protocole augmentés du comportement GTSM devront être utilisés si la négociation dynamique est estimée nécessaire.

2.2 Hypothèses sur la sophistication des attaques

Dans le présent document, on suppose que les attaquants potentiels ont évolué à la fois en sophistication et dans l'accès au point qu'ils peuvent envoyer du trafic de contrôle à une session de protocole, et que ce trafic paraît être du trafic de contrôle valide (c'est-à-dire, il a la source/destination de routeurs homologues configurés).

On suppose aussi que chaque routeur sur le chemin entre l'attaquant et le locuteur de protocole victime décrémente correctement le TTL (en clair, si le chemin ou l'homologue adjacent est compromis, il y a alors des problèmes bien pires à résoudre).

Pour une protection maximale, le filtrage d'entrée devrait être appliqué avant que le paquet passe par la ressource rare. Autrement un attaquant directement connecté à une interface pourrait perturber une session protégée par GTSM sur la même interface ou une autre. Les interfaces qui ne sont pas configurées avec ce filtrage (par exemple, les liaisons de cœur de réseau) sont supposées ne pas avoir de tels attaquants (c'est-à-dire, elles sont de confiance).

Comme instance spécifique de telles interfaces, on suppose que les tunnels ne sont pas une porte dérobée pour permettre l'usurpation de TTL sur les paquets de protocole d'une session d'échange de trafic protégée par GTSM avec un voisin directement connecté. On suppose que : 1) il n'y a pas de paquets tunnelés qui se terminent sur le routeur, 2) les tunnels qui se terminent au routeur sont supposés être sûrs et les points d'extrémité sont de confiance, 3) la désencapsulation de tunnel inclut la prévention de l'usurpation de l'adresse de source [RFC3704], ou 4) la session à capacité GTSM ne permet pas que les paquets de protocole viennent d'un tunnel.

Comme la vaste majorité des échanges de trafic est entre des routeurs adjacents, on peut régler le TTL sur les paquets de protocole à 255 (le maximum possible pour IP) et ensuite rejeter tous les paquets de protocole qui viennent d'homologues configurés qui n'ont PAS un TTL entrant de 255.

GTSM peut être désactivé pour des applications comme les serveurs d'acheminement et autres serveurs d'échange de trafic multi bonds. Au cas où une attaque viendrait d'un échange de trafic multi bonds compromis, cet échange de trafic peut être fermé.

3. Procédure GTSM

Si GTSM n'est pas incorporé dans le protocole et est utilisé comme caractéristique supplémentaire (par exemple, pour BGP, LDP, ou MSDP) il NE DEVRAIT PAS être activé par défaut afin de rester rétro compatible avec le protocole non modifié. Cependant, si le protocole définit une négociation de capacité dynamique pour GTSM, un homologue de protocole PEUT suggérer l'utilisation de GTSM pourvu que GTSM ne soit activé que si les deux homologues s'accordent pour l'utiliser.

Si GTSM est activé pour une session de protocole, les étapes suivantes sont ajoutées aux procédures d'envoi et de réception de paquet IP :

Envoi des paquets de protocole :
Le champ TTL dans tous les paquets IP utilisés pour la transmission des messages associés aux sessions de protocoles avec GTSM activé DOIT être réglé à 255. Cela s'applique aussi aux messages ICMP de traitement d'erreur qui s'y rapportent.

Dans certaines architectures, le TTL du trafic généré par le plan de contrôle est dans certaines configurations décrémenté dans le plan de transmission. Le TTL des sessions à capacité GTSM NE DOIT PAS être décrémenté.

Réception des paquets de protocole :
L'étape d'identification du paquet GTSM associe chaque paquet reçu adressé au plan de contrôle du routeur à une des trois catégories de confiance suivantes :

+ Inconnue : ce sont des paquets qui ne peuvent pas être associés à une session enregistrée à capacité GTSM, et donc GTSM ne peut pas porter de jugement sur le niveau de risque qui leur est associé.
+ De confiance : ce sont des paquets qui ont été identifiés comme appartenant à une des sessions à capacité GTSM, et leurs valeurs de TTL sont dans la gamme attendue.
+ Dangereux : ce sont des paquets qui ont été identifiés comme appartenant à une des sessions à capacité GTSM, mais leurs valeurs de TTL ne sont PAS dans la gamme attendue, et donc GTSM estime qu'il y a un risque que ces paquets aient été falsifiés.

Les politiques exactes appliquées aux paquets des différentes classes ne sont pas spécifiées dans le présent document et il est attendu qu'elles soient configurables. La configurabilité est probablement nécessaire en particulier avec le traitement des

messages qui s'y rapportent (erreurs ICMP). On devrait noter que la fragmentation peut restreindre la quantité d'informations disponible pour la classification.

Cependant, par défaut, les mises en œuvre ,:
+ ,DEVRAIENT s'assurer que les paquets classés comme dangereux ne rentrent pas en compétition pour les ressources avec les paquets classés comme de confiance ou inconnus.
+ NE DOIVENT PAS éliminer (au titre du traitement de GTSM) des paquets classés comme de confiance ou inconnus.
+ PEUVENT éliminer les paquets classés comme dangereux.

4. Remerciements

L'utilisation du champ TTL pour protéger BGP a pour origine de nombreuses personnes différentes, parmi lesquelles Paul Traina et Jon Stewart. Ryan McDowell a aussi suggéré une idée similaire. Steve Bellovin, Jay Borkenhagen, Randy Bush, Alfred Hoenes, Vern Paxon, Robert Raszuk, et Alex Zinin ont aussi fourni des retours utiles sur les versions antérieures de ce document. David Ward a fourni des conseils sur la généralisation de l'idée originale spécifique de BGP. Alex Zinin, Alia Atlas, et John Scudder ont fourni une quantité significative de réactions sur les plus récentes versions du document. Durant et après le dernier appel de l'IETF, des commentaires utiles ont été fournis par Francis Dupont, Sam Hartman, Lars Eggert, et Ross Callon.

5. Considérations pour la sécurité

GTSM est une simple procédure qui protège les sessions de protocole d'un seul bond, sauf dans les cas où l'homologue a été compromis. En particulier, elle ne protège pas contre la large gamme d'attaques sur le réseau ; la protection contre ces attaques exige des mécanismes de sécurité plus rigoureux.

5.1 Falsification de TTL (limite de bonds)

L'approche décrite ici se fonde sur l'observation qu'une valeur de TTL (ou limite de bonds) de 255 n'est pas triviale à falsifier, car lorsque le paquet passe de routeur en routeur jusqu'à sa destination, le TTL est décrémenté de un à chaque routeur. Par suite, quand un routeur reçoit un paquet, il peut n'être pas capable de déterminer si l'adresse IP du paquet est valide, mais il peut déterminer à combien de bonds de routeur il est (là encore, on suppose qu'aucun des routeurs sur le chemin n'est compromis d'une façon telle qu'il réinitialiserait le TTL du paquet).

Noter, cependant, que bien que la manipulation du TTL d'un paquet de telle façon qu'il ait une valeur particulière quand il est originaire d'une localisation arbitraire soit difficile (mais pas impossible) générer une valeur de TTL de 255 à partir de localisations non directement connectées n'est pas possible (là encore, en supposant qu'aucun des voisins directement connectés n'est compromis, que le paquet n'a pas été tunnelé au désencapsuleur, et que les routeurs intermédiaires fonctionnent en accord avec la [RFC0791]).

5.2 Paquets tunnelés

La sécurité de toute technique de tunnelage dépend largement de l'authentification aux points d'extrémité du tunnel, ainsi que de la façon dont les paquets tunnelés sont protégés en vol. De tels mécanismes sortent cependant du domaine d'application du présent mémoire.

Une exception à l'observation qu'un paquet avec un TTL de 255 est difficilte à contrefaire peut survenir quand un paquet de protocole est tunnelé et que le tunnel n'est pas protégé en intégrité (c'est-à-dire, que la couche inférieure est compromise).

Quand le paquet de protocole est tunnelé directement à l'homologue de protocole (c'est-à-dire, l'homologue de protocole est le désencapsuleur) le GTSM fournit un surplus limité de protection car la sécurité dépend entièrement de l'intégrité du tunnel.

Pour les adjacences de protocole sur un tunnel, si le tunnel lui-même est réputé sûr (c'est-à-dire, si l'infrastructure sous-jacente est réputée sûre, et si le tunnel offre des degrés de protection contre l'usurpation tels que des clés ou un chiffrement) le GTSM peut servir de vérification que le paquet de protocole ne provient pas d'au delà de la tête du tunnel. De plus, si l'homologue de protocole peut recevoir des paquets pour la session de protocole protégée par GTSM provenant de l'extérieur du tunnel, le GTSM peut aider à déjouer des attaques provenant d'au delà du routeur adjacent.

Quand l'extrémité de queue du tunnel désencapsule le paquet de protocole et ensuite le transmet sur IP à un homologue de protocole directement connecté, le TTL est décrémenté comme décrit ci-dessous. Cela signifie que le désencapsuleur du tunnel est l'avant dernier nœud du point de vue de l'homologue de protocole protégé par GTSM. Par suite, la vérification GTSM protège contre des attaquants qui encapsuleraient des paquets pour vos homologues. Cependant, il y a des cas spécifiques où la connexion du nœud désencapsuleur de tunnel à l'homologue de protocole n'est pas un bond de transmission IP, où la diminution du TTL n'intervient pas (par exemple, un tunnelage de couche, un pontage, etc). Dans l'architecture IPsec [RFC4301], un autre exemple est l'utilisation de la prise sur le réseau (BITW, *Bump-in-the-Wire*) [BITW].

5.2.1 IP tunnelé sur IP

Les paquets de protocole peuvent être tunnelés sur IP directement à un homologue de protocole, ou à un désencapsuleur (point d'extrémité de tunnel) qui transmet alors le paquet à un homologue de protocole directement connecté. Les exemples de tunnelage IP sur IP incluent IP dans IP [RFC2003], GRE [RFC2784], et diverses formes de IPv6 dans IPv4 (par exemple, [RFC4213]). Ces cas sont décrits ci-dessous.

Routeur homologue ---------- Routeur de point d'extrémité de tunnel et homologue
TTL=255 [tunnel] [TTL=255 à l'entrée] [TTL=255 au traitement]

Routeur homologue ---------- Routeur de point d'extrémité de tunnel ----- Homologue sur la liaison
TTL=255 [tunnel] [TTL=255 à l'entrée] [TTL=254 à l'entrée] [TTL=254 à la sortie]

Dans les deux cas, l'encapsulateur (point d'extrémité d'origine de tunnel) est l'homologue (supposé) de protocole d'envoi. Le TTL dans le datagramme IP interne peut être réglé à 255, car la RFC 2003 spécifie le comportement suivant :

Lors de l'encapsulation d'un datagramme, le TTL dans l'en-tête IP interne est décrémenté de un si le tunnelage est fait au titre de la transmission du datagramme ; autrement, le TTL de l'en-tête interne n'est pas changé durant l'encapsulation.

Dans le premier cas, le paquet encapsulé est tunnelé directement à l'homologue de protocole (qui est aussi un point d'extrémité de tunnel) et donc le TL du paquet encapsulé peut être reçu par l'homologue de protocole avec une valeur arbitraire, incluant 255.

Dans le second cas, le paquet encapsulé est tunnelé à un désencapsuleur (point d'extrémité de tunnel) qui le transmet alors à un homologue de protocole directement connecté. Pour les tunnels IP dans IP, la RFC 2003 spécifie le comportement de désencapsuleur suivant :

Le TTL dans l'en-tête IP interne n'est pas changé à la désencapsulation. Si, après la désencapsulation, le datagramme interne a un TTL = 0, le désencapsuleur DOIT éliminer le datagramme. Si, après la désencapsulation, le désencapsuleur transmet le datagramme à une de ses interfaces réseau, il va décrémenter le TTL par suite d'une transmission IP normale. Voir aussi le paragraphe 4.4.

Et similairement, pour les tunnels GRE, la RFC 2784 spécifie le comportement de désencapsuleur suivant : quand un point d'extrémité de tunnel désencapsule un paquet GRE qui a un paquet IPv4 comme charge utile, l'adresse de destination dans l'en-tête du paquet de charge utile IPv4 DOIT être utilisée pour transmettre le paquet et le TTL du paquet de charge utile DOIT être décrémenté.

Donc, le TTL de l'en-tête du paquet IP interne, tel que vu par le désencapsuleur, peut être réglé à une valeur arbitraire (en particulier, 255). Si le désencapsuleur est aussi l'homologue de protocole, il est possible de lui livrer le paquet de protocole avec un TTL de 255 (premier cas). Par ailleurs, si le désencapsuleur a besoin de transmettre le paquet de protocole à un homologue directement connecté, le TTL va être décrémenté (second cas).

5.2.2 IP tunnelé sur MPLS

Les paquets de protocole peuvent aussi être tunnelés sur des chemins de commutation d'étiquettes (LSP, *Label Switched Path*) MPLS à un homologue de protocole. Le diagramme qui suit décrit la topologie.

Routeur homologue ---------- Routeur de termination de LSP et homologue
TTL=255 LSP MPLS [TTL=x à l'entrée]

Les LSP MPLS peuvent fonctionner dans des modèles de tunnelage uniforme ou de tuyau. Le traitement du TTL pour ces modèles est décrit dans la [RFC3443] qui met à jour la [RFC3032] à l'égard du traitement de TTL dans les réseaux MPLS. La RFC 3443 spécifie le traitement de TTL dans les deux modèles, uniforme et de tuyau, qui à leur tour peuvent être utilisés avec ou sans saut de l'avant dernier bond (PHP, *penultimate hop popping*). Le traitement de TTL dans ces cas résulte en différents comportements, et sont donc analysés séparément. Voir les paragraphes 3.1 à 3.3 de la RFC 3443.

La principale différence entre les modèles uniforme et de tuyau en matière de traitement de TTL au nœud de terminaison de LSP réside dans la façon de déterminer le TTL entrant (iTTL, *incoming TTL*). Le modèle de tunnelage détermine le iTTL : pour les LSP de modèle uniforme, le iTTL est la valeur du champ TTL provenant de l'en-tête MPLS sauté (en-tête encapsulant) tandis que pour les LSP de modèle tuyau, le iTTL est la valeur du champ TTL provenant de l'en-tête exposé (en-tête encapsulé).

Pour les LSP de modèle uniforme, la RFC 3443 déclare que à l'entrée : pour chaque étiquette poussée de modèle uniforme, le TTL est copié de l'étiquette/paquet IP immédiatement en dessous d'elle.

À partir de là, le TTL interne (c'est-à-dire, le TTL du datagramme IP tunnelé) représente une information non significative, et au nœud de sortie ou, durant le PHP, le iTTL est égal au TTL de l'en-tête MPLS sauté (voir le paragraphe 3.1 de la RFC 3443). En conséquence, pour les LSP de modèle uniforme de plus d'un bond, le TTL à l'entrée (iTTL) va être de moins que 255 ($x \leq 254$) et par suite, la vérification décrite à la Section 3 du présent document va échouer.

Le traitement de TTL est identique dans les LSP de modèle à tuyau court sans PHP et les LSP de modèle tuyau (seulement sans PHP). Pour ces deux cas, la RFC 3443 déclare que : pour chaque étiquette poussée de modèle tuyau ou de modèle tuyau court, le champ TTL est réglé à une valeur configurée par l'opérateur du réseau. Dans la plupart des mises en œuvre, cette valeur est réglée à 255 par défaut.

Dans ces modèles, le traitement de transmission à la sortie se fonde sur le paquet tunnelé par opposition à l'encapsulation de paquet. Le TTL d'entrée (iTTL) est la valeur du champ TTL de l'en-tête qui est exposé, c'est-à-dire le TTL du datagramme IP tunnelé. Le TTL du paquet de protocole tel que vu par la terminaison du LSP peut donc être réglé à une valeur arbitraire (incluant 255). Si le routeur de terminaison du LSP est aussi l'homologue de protocole, il est possible de livrer le paquet de protocole avec un TTL de 255 ($x = 255$).

Finalement, pour les LSP du modèle de tuyau court avec PHP, le TTL du paquet tunnelé est inchangé après l'opération de PHP. Donc, les mêmes conclusions que pour les LSP du modèle de tuyau court sans PHP et du modèle de tuyau (seulement sans PHP) s'appliquent à ce cas. Pour les LSP du modèle à tuyau court, le TTL à la sortie a la même valeur avec ou sans PHP.

En conclusion, les vérifications GTSM sont possibles pour IP tunnelé sur des LSP de modèle tuyau, mais pas pour IP tunnelé sur des LSP de modèle uniforme. De plus, pour tous les modes de tunnelages, si le routeur de terminaison de LSP a besoin de transmettre le paquet de protocole à un homologue de protocole directement connecté, il n'est pas possible de livrer le paquet de protocole à l'homologue de protocole avec un TTL de 255. Si le paquet est retransmis plus loin, le TTL sortant (oTTL, *outgoing TTL*) est calculé en décrémentant le iTTL de un.

5.3 Attaques sur la liaison

Comme décrit à la Section 2, un attaquant directement connecté à une interface peu perturber une session protégée par GTSM sur la même interface ou une autre (en falsifiant l'adresse d'un homologue GTSM) sauf si le filtrage d'entrée a été appliqué à l'interface de connexion. Par suite, les interfaces qui ne comportent pas une telle protection doivent être de confiance pour ne pas générer d'attaques contre le routeur.

5.4 Considérations sur la fragmentation

Comme mentionné précédemment, la fragmentation peut restreindre la quantité d'informations disponibles pour la classification. Comme les fragments IP non initiaux ne contiennent pas d'informations de couche 4, il est très probable qu'ils ne peuvent pas être associés à une session à capacité GTSM enregistrée. Suivant les procédures de protocole de réception décrites à la Section 3, les fragments IP non initiaux vont probablement être classés avec le niveau de confiance Inconnu. Et comme le paquet IP va devoir être réassemblé afin d'être traité, le résultat final est que le fragment initial d'une session à capacité GTSM reçoit effectivement le traitement d'un paquet de niveau de confiance inconnu, et le paquet réassemblé complet reçoit l'agrégation des inconnus.

En principe, une mise en œuvre pourrait se souvenir du TTL de tous les fragments reçus. Puis, lors du réassemblage du paquet, vérifier que le TTL de tous les fragments correspond à la valeur requise pour une session associée à capacité GTSM. Dans le cas probablement courant où la mise en œuvre ne fait pas cette vérification sur tous les fragments, il est alors possible qu'un premier fragment légitime (qui réussi la vérification GTSM) soit combiné à des fragments non initiaux falsifiés, ce qui implique que l'intégrité du paquet reçu sera inconnue et non protégée. Si cette vérification est effectuée sur tous les fragments au réassemblage, et si certains fragment ne réussissent pas à la vérification GTSM pour une session à capacité GTSM, le paquet réassemblé est rangé dans la catégorie de niveau de confiance dangereux et reçoit le traitement correspondant.

De plus, le réassemblage exige d'attendre d'avoir tous les fragments et donc invalide ou affaiblit probablement la cinquième hypothèse présentée à la Section 2 : il peut n'être pas possible de classer les fragments non initiaux avant d'aller plus loin sur une ressource rare dans le système, quand les fragments doivent être mis en mémoire tampon pour le réassemblage et ensuite être traités par une CPU. C'est-à-dire, quand le classement ne peut pas être fait avec la granularité requise, les fragments non initiaux des paquets d'une session à capacité GTSM n'utiliseraient pas des réservoirs de ressources différents.

Par conséquent, pour obtenir une protection pratique contre les attaques de fragments, les opérateurs peuvent devoir limiter en débit ou éliminer tous les fragments reçus. À ce titre, il est fortement RECOMMANDÉ que les protocoles protégés par GTSM évitent la fragmentation et le réassemblage par un réglage manuel de la MTU, en utilisant des mesures d'adaptation telles que la découverte de la MTU de chemin (PMTUD, *Path MTU Discovery*) ou toute autre méthode disponible des [RFC1191], [RFC1981], ou [RFC4821].

5.5 Sessions de protocole multi bonds

GTSM pourrait éventuellement offrir un certain petit, bien que difficile à quantifier, degré de protection lorsque utilisé avec des sessions multi-bonds (voir l'Appendice A). Afin d'éviter d'avoir à quantifier le degré de protection et l'applicabilité résultante du multi-bonds, on décrit seulement le cas d'un seul bond parce que ses propriétés de sécurité sont plus claires.

6. Déclaration d'applicabilité

GTSM est seulement applicable aux environnements dont la topologie est limitée par nature (et est plus efficace dans les cas où les homologues de protocole sont directement connectés). En particulier, son application devrait être limitée aux cas où les homologues de protocole sont directement connectés.

GTSM ne va pas protéger contre des attaquants qui sont proches de la station protégée comme son homologue légitime. Par exemple, si l'homologue légitime est un bond plus loin, GTSM ne protègera pas des attaques d'appareils directement connectés sur la même interface (en voir plus au paragraphe 2.2).

Une expérimentation sur l'applicabilité et les propriétés de GTSM est nécessaire dans les scénarios multi-bonds. Les scénarios multi-bonds où GTSM pourrait être applicable sont supposés avoir les caractéristiques suivantes : la topologie entre les homologues est très statique et bien connue, et le réseau intermédiaire (entre les homologues) est de confiance.

6.1 Rétro compatibilité

La [RFC3682] ne spécifiait pas comment traiter les "messages en relation" (erreurs ICMP). La présente spécification rend obligatoire le réglage et la vérification de TTL=255 sur ces messages ainsi que que les paquets du protocole principal.

Régler TTL=255 dans les messages en relation ne pose pas de problème pour les mises en œuvre de la RFC 3682.

Exiger TTL=255 dans les messages en relation peut avoir un impact sur les mises en œuvre de la RFC 3682, selon le TTL que la mise en œuvre utilise par défaut pour les paquets générés ; certaines mises en œuvre sont connues pour utiliser 255, tandis que 64 ou d'autres valeurs sont aussi utilisées. Les messages en relation provenant de la dernière catégorie des mises en œuvre de la RFC 3682 seraient classés comme dangereux et traités comme décrit à la Section 3. On estime que cela ne constitue pas un problème significatif parce que les protocoles ne dépendent pas des messages en relation (par exemple, ayant normalement un échange de protocole pour clore la session au lieu de faire un TCP-RST) et bien sûr, la livraison des messages en relation n'est pas fiable. À ce titre, les messages en relation fournissent normalement une optimisation pour raccourcir la temporisation de garde en vie d'un protocole. Sans considération de ces problèmes, étant donné que les

messages en relation fournissent un vecteur d'attaque significatif pour, par exemple, réinitialiser les sessions de protocole, il y a un sens à ajouter cette restriction.

7. Références

7.1 Références normatives

[RFC0791] J. Postel, éd., "Protocole Internet - Spécification du protocole du programme Internet", STD 5, septembre 1981.

[RFC2003] C. Perkins, "Encapsulation de IP dans IP", octobre 1996. (*MàJ par* RFC 3168, RFC 6864, Errata) *(P.S.)*

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

[RFC2461] T. Narten, E. Nordmark, W. Simpson, "Découverte de voisins pour IP version 6 (IPv6)", décembre 1998. (*Obsolète, voir* RFC4861) *(D.S.)*

[RFC2784] D. Farinacci, T. Li, S. Hanks, D. Meyer et P. Traina, "Encapsulation d'acheminement générique (GRE)", mars 2000.

[RFC3392] R. Chandra et J. Scudder, "Annonces de capacités avec BGP-4", novembre 2002. *(Obsolète, voir* RFC5492)

[RFC3443] P. Agarwal, B. Akyol, "Traitement de la durée de vie (TTL) dans les réseaux à commutation d'étiquettes multi-protocoles (MPLS)", janvier 2003. *(P.S.)*

[RFC4213] E. Nordmark, R. Gilligan, "Mécanismes de transition de base pour hôtes et routeurs IPv6", octobre 2005. *(P.S.)*

[RFC4271] Y. Rekhter, T. Li et S. Hares, "Protocole de routeur frontière version 4 (BGP-4)", janvier 2006. *(D.S.) (MàJ par* RFC6608, RFC8212)

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

7.2 Références pour information

[BITW] "Thread: 'IP-in-IP, TTL decrementing when forwarding and BITW' on int-area list, Message-ID: <Pine.LNX.4.64.0606020830220.12705@netcore.fi>", juin 2006, <http://www1.ietf.org/mail-archive/web/int-area/current/msg00267.html>.

[RFC1191] J. Mogul et S. Deering, "Découverte de la MTU de chemin", novembre 1990.

[RFC1981] J. McCann, S. Deering, J. Mogul, "Découverte de la MTU de chemin pour IP version 6", août 1996. *(D.S. ; Remplacé par* [RFC8201], STD87)

[RFC3032] E. Rosen et autres, "Codage de pile d'étiquettes MPLS", janvier 2001.

[RFC3682] V. Gill, J. Heasley, D. Meyer, "Mécanisme TTL de sécurité généralisé (GTSM)", février 2004. (*Obsolète, voir* RFC5082) *(Expérimentale)*

[RFC3704] F. Baker, P. Savola, "Filtrage d'entrée pour réseaux à rattachement multiples", mars 2004. (BCP0084) (*MàJ par* RFC8704)

[RFC4272] S. Murphy, "Analyse des faiblesses de la sécurité de BGP", janvier 2006. *(Information)*

[RFC4821] M. Mathis, J. Heffner, "Découverte de la MTU de chemin de couche de mise en paquet", mars 2007. *(P.S.)*

Appendice A. GTSM multi bonds

Note : Cette partie de la spécification n'est pas normative.

La principale application de GTSM est pour les homologues directement connectés. GTSM pourrait aussi être utilisé pour des sessions non directement connectées, où le receveur vérifierait que le TTL est dans un nombre configuré de bonds jusqu'à 255 (par exemple, vérifier que les paquets ont 254 ou 255). Comme un tel déploiement est supposé avoir une applicabilité plus limitée et des implications de sécurité différentes, il n'est pas spécifié dans le présent document.

Appendice B. Changements par rapport à la RFC 3682

- Amener le document sur la voie de la normalisation (la RFC 3682 était expérimentale).
- Nouveau texte sur l'applicabilité de GTSM et utilisation dans des protocoles nouveaux et existants.
- Restriction de la portée à ne pas spécifier les scénarios multi-bonds.
- Exiger explicitement que les messages en rapport (erreurs ICMP) doivent aussi être envoyés et vérifiés comme ayant un TTL=255. Voir au paragraphe 6.1 la discussion sur la rétro compatibilité.
- Précisions relatives à la fragmentation, la sécurité avec le tunnelage, et les implications sur le filtrage à l'entrée.
- Un nombre significatif d'améliorations et précisions rédactionnelles.

Adresse des auteurs

Vijay Gill
mél : mailto:vijay@umbc.edu

John Heasley
mél : heas@shrubbery.net

David Meyer
mél : dmm@1-4-5.net

Carlos Pignataro
mél : cpignata@cisco.com

Pekka Savola (éditeur)
Espoo
Finland
mél : psavola@funet.fi

Déclaration complète des droits de reproduction

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 la Internet Society.