Fonctionnement des Keepalives GRE

Contenu

Introduction

Ce document fournit un aperçu de la façon dont fonctionnent les keepalives GRE (Generic Routing Encapsulation).

Conditions préalables

Conditions requises

Les lecteurs de ce document devraient avoir connaissance des sujets suivants :

- Maintien en vie du tunnel GRE
- Commandes du mode de configuration Keepalive

Components Used

Les informations contenues dans ce document sont basées sur les versions de matériel et de logiciel suivantes :

- Routeur Cisco 7505
- Logiciel Cisco IOS® prenant en charge GRE sur IPSec

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, make sure that you understand the potential impact of any command.

Conventions

For more information on document conventions, refer to the Cisco Technical Tips Conventions.

Informations générales

La fonctionnalité keepalive GRE active la commande d'interface **keepalive** pour les tunnels et vous permet de configurer des keepalives pour les tunnels GRE point à point. Vous pouvez configurer les keepalives à l'aide de la commande **keepalive**, et éventuellement avec sa nouvelle extension.

Les tunnels GRE fournissent une méthode pour encapsuler des paquets arbitraires à l'intérieur d'un protocole de transport. Ils offrent également une architecture conçue pour fournir les services nécessaires à la mise en oeuvre de tout schéma d'encapsulation point à point standard. Voici quelques-uns des avantages des tunnels GRE :

- Les tunnels GRE fournissent des réseaux locaux multiprotocoles sur un fédérateur à protocole unique.
- Les tunnels GRE fournissent des solutions de contournement pour les réseaux qui contiennent des protocoles avec un nombre de sauts limité.
- Les tunnels GRE connectent des sous-réseaux discontinus.
- Les tunnels GRE autorisent les VPN sur les WAN.

Cependant, dans l'implémentation actuelle des tunnels GRE, un tunnel configuré n'a pas la capacité de mettre hors service le protocole de ligne de l'un ou l'autre point de terminaison de tunnel, si l'extrémité distante est inaccessible. Ainsi, le trafic envoyé à partir du tunnel est creusé en noir, et il ne peut pas suivre d'autres chemins car le tunnel reste toujours actif.

Cette situation est vraie pour les tunnels qui reposent sur des routes statiques ou sur des protocoles de routage qui regroupent des routes pour trouver une route vers la destination du tunnel. Cela est également vrai dans les situations où les données du plan de contrôle suivent un chemin différent des données du plan de données.

Le mécanisme de conservation du tunnel

Cette section fournit une description fonctionnelle du mécanisme de keepalive de tunnel à l'aide d'un exemple. Cette section répertorie également les éléments logiciels modifiés par cette fonctionnalité et décrit l'impact sur la mémoire et les performances.

Description fonctionnelle

Le mécanisme de keepalive de tunnel active, étend et implémente une commande spécifique à l'interface pour les interfaces de tunnel, et offre la possibilité de mettre hors service le protocole de ligne d'un tunnel. Pour plus d'informations, consultez la section Commandes et configuration.

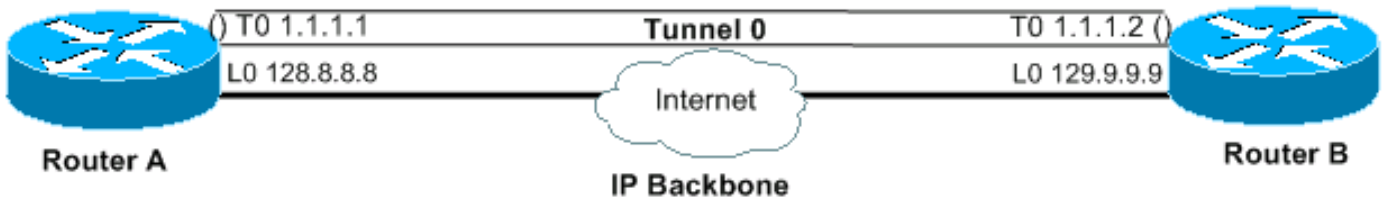
Le mécanisme de keepalive du tunnel répond également à ces exigences supplémentaires :

- Le mécanisme de keepalive du tunnel fonctionne même si le point de terminaison du tunnel distant ne prend pas en charge les keepalives.
- Le mécanisme de keepalive du tunnel génère des keepalives.
- Le mécanisme de keepalive du tunnel traite les keepalives.

- Le mécanisme de keepalive du tunnel répond aux paquets keepalive de l'extrémité distante, même lorsque le protocole de ligne du tunnel est désactivé.

Voici un exemple du fonctionnement du mécanisme de keepalive du tunnel (voir Figure 1) :

Figure 1 - Exemple de mécanisme de maintien du tunnel



Sortie

```
interface tunnel 0                          interface tunnel 0
ip address 1.1.1.1 255.255.255.240          ip address 1.1.1.2 255.255.255.240
tunnel source 128.8.8.8                     tunnel source 129.9.9.9
tunnel destination 129.9.9.9                tunnel destination 128.8.8.8
keepalive 5 4                               keepalive 5 4
interface loopback 0                        interface loopback 0
ip address 128.8.8.8 255.255.255.255       ip address 129.9.9.9 255.255.255.255
```

Paquet keepalive provenant de A à B

```
      ---outer IP header---'      ---inner IP header---'
      ============================================================
      |IP | IP src  | IP dst  | GRE | IP | IP src  | IP dst  | GRE |
      |   |128.8.8.8|129.9.9.9|PT=IP|    |129.9.9.9|128.8.8.8| PT=0|
      ============================================================
                              ----'                         ---'
                               GRE header                    GRE header
```

Lorsque vous activez les keepalives sur le point d'extrémité du tunnel du routeur A, le routeur construit l'en-tête IP interne à chaque intervalle. À la fin de l'en-tête, le routeur ajoute également un en-tête GRE avec un type de protocole (PT) de 0, et aucune autre charge utile. Le routeur envoie ensuite ce paquet via le tunnel, ce qui entraîne son encapsulation avec l'en-tête IP externe et un en-tête GRE avec le PT d'IP. Le compteur de keepalive du tunnel s'incrémente d'un. S'il existe un moyen d'atteindre le point de terminaison du tunnel de l'extrémité distante et que le protocole de ligne de tunnel n'est pas désactivé pour d'autres raisons, le paquet arrive sur le routeur B. Il est ensuite mis en correspondance avec le tunnel 0, décapsulé et transféré à l'adresse IP de destination, qui est la source du tunnel, le routeur A. À l'arrivée sur le routeur A, le paquet est à nouveau décapsulé et le PT est vérifié. Si le résultat de la vérification PT est 0, cela signifie qu'il s'agit d'un paquet keepalive. Dans ce cas, le compteur de keepalive du tunnel est réinitialisé à 0 et le paquet est rejeté.

Si le routeur B est inaccessible, le routeur A continue à construire et à envoyer les paquets keepalive avec le trafic normal. Si le protocole de ligne est désactivé, les keepalives ne reviennent pas au routeur A. Par conséquent, le compteur de keepalive continue d'augmenter. Le protocole de ligne de tunnel reste actif uniquement tant que le compteur de keepalive de tunnel reste zéro ou inférieur à une valeur configurée. Si cette condition n'est pas vraie, lors de la prochaine tentative d'envoi d'un keepalive au routeur B, le protocole de ligne est désactivé, dès que le compteur keepalive atteint la valeur keepalive configurée. Dans l'état up/down, le tunnel ne transfère ni ne traite aucun trafic en dehors des paquets keepalive. Pour que cela fonctionne uniquement pour les paquets de test d'activité, le tunnel doit être convivial pour le transfert et la

réception. Ainsi, l'algorithme de recherche de tunnel doit réussir dans tous les cas et ne doit ignorer que les paquets de données si le protocole de ligne est désactivé. Lorsqu'un paquet keepalive est reçu, cela signifie que le point de terminaison du tunnel est à nouveau accessible. Le compteur de keepalive du tunnel est ensuite réinitialisé à 0 et le protocole de ligne est rétabli.

Impact sur la mémoire et les performances

Cette fonctionnalité ne nécessite pratiquement aucune demande supplémentaire sur la mémoire système du routeur et les performances ne devraient pas être affectées par son ajout. Les paquets Keepalive sont traités comme des paquets ordinaires, et il est donc possible qu'ils puissent être abandonnés dans des conditions de trafic élevées. Pour l'instant, vous pouvez modifier le nombre de tentatives pour traiter ce problème. Si cela s'avère finalement insuffisant, vous pouvez placer les paquets keepalive générés localement dans une file d'attente de priorité élevée pour la transmission. Vous pouvez ensuite définir la valeur TOS dans les en-têtes IP sur une valeur plus appropriée, autre que la valeur par défaut ou configurée.

Considérations relatives à l'emballage

Cette fonctionnalité est incluse dans le code de tunnel IP de base et dans le sous-système GRE. Par conséquent, il doit être disponible avec un paquet IP de base qui a le tunnel et les sous-systèmes GRE.

Commandes et configuration

Cette section traite de la commande **keepalive** activée et étendue par cette fonctionnalité uniquement sous l'ID de bogue Cisco CSCuk26449. D'autres commandes sont documentées dans les *guides de configuration et les références de commandes de Cisco IOS respectifs*. La commande **[no] keepalive <*période*> <*retries*>** est activée et étendue avec un deuxième paramètre et est disponible dans le logiciel Cisco IOS Version 12.2(8)T et ultérieure. Il a également été porté sous l'ID de bogue Cisco CSCuk29980 et CSCuk29983 aux versions 12.1E et 12.2S du logiciel Cisco IOS.

Comme **keepalive** est une commande de configuration d'interface qui active les keepalives sur l'interface de tunnel, seuls les keepalives pour le mode GRE/IP sont actuellement pris en charge. Le deuxième paramètre de la commande (*retries*) est visible et disponible uniquement pour les interfaces de tunnel. Les valeurs du premier paramètre peuvent être comprises entre 1 et 32 767. Lorsque la valeur est 0, elle équivaut à « no keepalive ». Ce paramètre a une valeur par défaut de 10. Les valeurs du second paramètre peuvent être comprises entre 1 et 255, et il indique le nombre de messages de test d'activité envoyés mais non retournés, après quoi l'interface de tunnel désactive le protocole de ligne. Les keepalives sur les interfaces de tunnel sont désactivés par défaut.

Exemples de formats de sortie et d'écran

Cette section fournit des exemples de résultats.

```
cisco-7505#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
cisco-7505(config)#interface tunnel 1
cisco-7505(config-if)#?
  access-expression   Build a bridge boolean access expression
```

```
  ..............
  keepalive           Enable keepalive<=====
  ..............
  timeout             Define timeout values for this interface

cisco-7505(config-if)#keepalive ?<=====
  <0-32767>  Keepalive period (default 10 seconds)

cisco-7505(config-if)#keepalive 5 ?<=====
  <1-255>    Keepalive retries (default 3 times)
cisco-7505(config-if)#keepalive 5 4<=====
cisco-7505(config-if)#end

cisco-7505#show interfaces tunnel 1

Tunnel1 is up, line protocol is up
  Hardware is Tunnel
  Internet address is 10.1.1.1/24
  MTU 1514 bytes, BW 9 Kbit, DLY 500000 usec, rely 255/255, load 1/255
  Encapsulation TUNNEL, loopback not set
  Keepalive set (5 sec), retries 4<=====
  Tunnel source 9.2.2.1, destination 6.6.6.2
  Tunnel protocol/transport GRE/IP, key disabled, sequencing disabled
  Tunnel TOS 0xF, Tunnel TTL 128
  Checksumming of packets disabled, fast tunneling enabled
  Last input never, output 00:57:05, output hang never
  Last clearing of "show interface" counters never
  Queueing strategy: fifo
  Output queue 0/0, 1 drops; input queue 0/75, 0 drops
  30 second input rate 0 bits/sec, 0 packets/sec
  30 second output rate 0 bits/sec, 0 packets/sec
     0 packets input, 0 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     3 packets output, 1860 bytes, 0 underruns
     0 output errors, 0 collisions, 0 interface resets
     0 output buffer failures, 0 output buffers swapped out
```

Informations connexes

- Keepalive de tunnel GRE (Generic Routing Encapsulation)
- Exemples de configuration GRE
- Assistance technique et documentation