GRE-tunnelhandleidingen begrijpen

Inhoud

[Inleiding](#)
[GRE-tunnels](#)
[Hoe Tunnel Keepalives werkt](#)
[GRE-tunnelkeepalives](#)
[GRE-keepalives en Unicast-doorsturen van omgekeerde paden](#)
[IPsec- en GRE-keepalives](#)
[GRE-tunnels met IPsec](#)
[Problemen met keepalives wanneer u IPsec en GRE combineert](#)
[Scenario 1](#)
[Scenario 2](#)
[Scenario 3](#)
[Tijdelijke oplossing](#)
[Gerelateerde informatie](#)

Inleiding

Dit document beschrijft wat de Generic Routing Encapsulation (GRE)-keepalives zijn en hoe ze werken.

GRE-tunnels

Een GRE-tunnel is een logische interface op een Cisco-router die een manier biedt om passagierspakketten in een transportprotocol in te kapselen. Het is een architectuur die is ontworpen om de diensten te verlenen om een point-to-point inkapselingsschema te implementeren.

GRE-tunnels zijn ontworpen om volledig stateloos te zijn. Dit betekent dat elk tunneleindpunt geen informatie houdt over de toestand of beschikbaarheid van het tunneleindpunt op afstand. Een gevolg van dit is dat de lokale router van het tunneleindpunt niet de capaciteit heeft om het lijnprotocol van de interface van de GRE Tunnel neer te brengen als het verre eind van de tunnel onbereikbaar is. De capaciteit om een interface zoals neer te merken wanneer het verre eind van de verbinding niet beschikbaar is wordt gebruikt om het even welke routes (specifiek statische routes) in de routeringstabel te verwijderen die die interface als uitgaande interface gebruiken. Specifiek, als het lijnprotocol voor een interface wordt veranderd in beneden, dan om het even welke statische routes die erop wijzen dat de interface wordt verwijderd uit de routeringstabel. Dit maakt de installatie van een alternatieve (zwevende) statische route of voor op beleid gebaseerde routing (PBR) mogelijk om een alternatieve next-hop of interface te selecteren.

Normaal gesproken komt een GRE Tunnel interface omhoog zodra deze is geconfigureerd en blijft omhoog zolang er een geldig tunnelbronadres of interface is die omhoog is. Het IP-adres van de tunnelbestemming moet ook routeerbaar zijn. Dit is waar zelfs als de andere kant van de tunnel niet is gevormd. Dit betekent dat een statische route of PBR-doorsturen van pakketten via de GRE-tunnelinterface van kracht blijft, ook al bereiken de GRE-tunnelpakketten niet het andere

uiteinde van de tunnel.

Voordat GRE keepalives werden geïmplementeerd, waren er alleen manieren om lokale problemen op de router te bepalen en geen manier om problemen in het tussenliggende netwerk te bepalen. Bijvoorbeeld, het geval waarin de GRE-tunnelpakketten met succes worden doorgestuurd, maar verloren gaan voordat ze de andere kant van de tunnel bereiken. Zulke scenario's zouden ervoor zorgen dat datapakketten die door de GRE-tunnel gaan "zwart gehold" zijn, ook al was er een alternatieve route die PBR of een zwevende statische route via een andere interface gebruikte. Keepalives op de GRE-tunnelinterface worden gebruikt om dit probleem op dezelfde manier op te lossen als keepalives worden gebruikt op fysieke interfaces.

> **Opmerking**: GRE-keepalives worden onder geen enkele omstandigheid ondersteund in combinatie met IPsec-tunnelbescherming. In dit document wordt deze kwestie besproken.

Hoe Tunnel Keepalives werkt

Het GRE tunnelkeepalive mechanisme is gelijkaardig aan PPP keepalives in zoverre dat het de capaciteit voor één kant geeft om keepalive pakketten te voortkomen en te ontvangen aan en van een verre router zelfs als de verre router geen GRE keepalives steunt. Aangezien GRE een pakkettunnelmechanisme is voor het tunnelen van IP binnen IP, kan een GRE IP tunnelpakket worden gebouwd binnen een ander GRE IP tunnelpakket. Voor GRE keepalives, de afzender bouwt het keepalive reactiepakket binnen het originele keepalive verzoekpakket voor zodat het verre eind slechts de standaarddecapsulation van GRE van de buitenste GRE IP kopbal moet doen en dan het binnenste IP GRE pakket aan de afzender moet terugkeren. Deze pakketten illustreren de concepten voor IP-tunneling waarbij GRE het inkapselingsprotocol is en IP het transportprotocol. Het passagiersprotocol is ook IP (hoewel het een ander protocol kan zijn zoals DECnet, Internetwork Packet Exchange (IPX) of Appletalk).

Normaal pakket:

IP-header	TCP-header	Telnet

Tunnelpakket:

GRE	IP-header	GRE	IP-header	TCP-header	Telnet

- IP is het transportprotocol.
- GRE is het inkapselingsprotocol.
- IP is het passagiersprotocol.

Hier is een voorbeeld van een keepalive pakket dat voortkomt uit router A en voor router B. bestemd is. De keepalive reactie die de router B aan router A terugkeert is reeds binnen de BinnenIP-Kop. Router B decapsuleert eenvoudig het keepalive pakket en verstuurt het terug uit de fysieke interface (S2). Het verwerkt het GRE keepalive-pakket net als elk ander GRE IP-datapakket.

GRE keepalives:

GRE IP-header		GRE		IP-header	GRE
Bron A	Bestemming	PT=IP	Bron B	Bestemming A	PT=0

B

Dit mechanisme veroorzaakt de keepalive reactie om de fysieke interface eerder dan de tunnelinterface door:sturen. Dit betekent dat het GRE keepalive reactiepakket niet wordt beïnvloed door enige uitvoerfuncties op de tunnelinterface, zoals 'tunnelbescherming ...', QoS, Virtual Routing and Forwarding (VRF), enzovoort.

> **Opmerking**: als een inkomende toegangscontrolelijst (ACL) op de GRE-tunnelinterface is geconfigureerd, moet het GRE-tunnelkeepalive-pakket worden toegestaan dat door het tegenovergestelde apparaat wordt verzonden. Als dit niet het geval is, wordt de GRE-tunnel van het tegenovergestelde apparaat neergehaald. (`**toegangslijst** <number> **vergunningsvrije host** <tunnelbron> **host** <tunnelbestemming>`)

Een ander attribuut van GRE tunnelkeepalives is dat de keepalive timers aan elke kant onafhankelijk zijn en niet moeten aanpassen, gelijkend op PPP keepalives.

> **Tip**: Het probleem met de configuratie van keepalives slechts aan één kant van de tunnel is dat alleen de router die keepalives heeft geconfigureerd zijn tunnelinterface markeert als de keepalive timer verloopt. De GRE-tunnelinterface aan de andere kant, waar keepalives niet zijn geconfigureerd, blijft omhoog zelfs als de andere kant van de tunnel is omlaag. De tunnel kan een zwart-gat worden voor pakketten die in de tunnel worden geleid van de kant die geen keepalives gevormd had.

> **Tip**: In een groot hub-and-spoke GRE-tunnelnetwerk kan het aangewezen zijn om GRE keepalives alleen te configureren aan de spaak kant en niet aan de hub kant. Dit is omdat het voor de spaak vaak belangrijker is om te ontdekken dat de hub onbereikbaar is en daarom switch naar een back-uppad (back-up voor bellen bijvoorbeeld).

GRE-tunnelkeepalives

Met Cisco IOS®-softwarerelease 12.2(8)T is het mogelijk om keepalives te configureren op een point-to-point GRE-tunnelinterface. Met deze verandering, sluit de tunnelinterface dynamisch af als keepalives voor een bepaalde periode ontbreken.

Raadpleeg voor meer informatie over de manier waarop andere vormen van keepalives werken Overzicht van Keepalive-mechanismen op Cisco IOS.

> **Opmerking**: GRE-tunnelkeepalives worden alleen ondersteund op point-to-point GRE-tunnels. Tunnel keepalives zijn configureerbaar op multipoint GRE (mGRE) tunnels maar hebben geen effect.

> **Opmerking**: In het algemeen kunnen tunnelkeepalives niet werken wanneer VRF's worden gebruikt op de tunnelinterface en de fVRF ("tunnel vrf ...") en iVRF ("ip Vrf-doorsturen ..." op tunnelinterface) niet overeenkomen. Dit is van cruciaal belang op het tunneleindpunt dat "de weerslag" vormt van het keeplive naar de aanvrager. Wanneer het keepalive verzoek wordt ontvangen wordt het ontvangen in fVRF en gedecapsuleerd. Dit onthult het vooraf gemaakte keepalive antwoord, dat dan naar de afzender moet worden teruggestuurd, MAAR dat het door:sturen in de context van iVRF op de tunnelinterface is. Daarom, als iVRF en fVRF niet

overeenkomen dan wordt het keepalive antwoordpakket niet teruggestuurd naar de afzender. Dit is ook het geval als u iVRF en/of fVRF vervangt door "global".

Deze uitvoer toont de opdrachten die u gebruikt om keepalives in GRE-tunnels te configureren.

```
Router#configure terminal
Router(config)#interface tunnel0
Router(config-if)#keepalive 5 4

!--- The syntax of this command is keepalive [seconds [retries]].



!--- Keepalives are sent every 5 seconds and 4 retries.
!--- Keepalives must be missed before the tunnel is shut down.
!--- The default values are 10 seconds for the interval and 3 retries.
```

Om beter te begrijpen hoe het tunnelkeepalive mechanisme werkt, overweeg deze voorbeeldtunneltopologie en configuratie:

Router A

```
interface loopback 0
ip address 192.168.1.1 255.255.255.255
interface tunnel 0
ip address 10.10.10.1 255.255.255.252
tunnel source loopback0
tunnel destination 192.168.1.2
keepalive 5 4
```

Router B

```
interface loopback 0
ip address 192.168.1.2 255.255.255.255
interface tunnel 0
ip address 10.10.10.2 255.255.255.252
tunnel source loopback0
tunnel destination 192.168.1.1
keepalive 5 4
```

In dit scenario voert router A de volgende stappen uit:

1. Construeert de interne IP-kop om de vijf seconden, waarbij:

   de bron is ingesteld als de lokale bestemming van de tunnel, die 192.168.1.2 isde bestemming is ingesteld als de lokale tunnelbron, te weten 192.168.1.1

   en er wordt een GRE-header toegevoegd met een Protocol Type (PT) van 0

   Pakket dat door router A wordt geproduceerd maar niet verzonden:

2. Verzendt dat pakket uit zijn tunnelinterface, die in de inkapseling van het pakket met de router IP kopbal resulteert waar:

   de bron is ingesteld als lokale de tunnelbron, die 192.168.1.1 isde bestemming is vastgesteld als de lokale tunnelbestemming, namelijk 192.168.1.2

   en er wordt een GRE-header toegevoegd met PT = IP.

   Packet dat van router A naar router B wordt verzonden:

3. Verhoogt de tunnel levend tegenovergesteld door één.

4. Met de veronderstelling dat er een manier is om het verre eindtunneleindpunt te bereiken en het protocol van de tunnellijn is niet beneden wegens andere redenen, komt het pakket op Router B. aan. Het wordt dan aangepast tegen Tunnel 0, wordt gedecapsuleerd, en door:sturen aan de bestemming IP die het tunnelbron IP adres op router A is.

   Verzonden van router B naar router A:

5. Bij aankomst op router A wordt het pakket gedecapsuleerd en de controle van de PT resulteert in 0. Dit betekent dat dit een keepalive pakket is. De tunnelkeepalive teller wordt dan teruggesteld aan 0 en het pakket wordt verworpen.

Als router B onbereikbaar is, blijft router A keepalive pakketten evenals normaal verkeer construeren en verzenden. Als de keepalives niet terugkomen, blijft het tunnellijnprotocol omhoog zolang de tunnelkeepalive teller minder is dan het aantal herhalingen, wat in dit geval vier is. Als die voorwaarde niet waar is, dan probeert de volgende keer router A om keepalive naar router B te verzenden, wordt het lijnprotocol neergehaald.

> **Opmerking**: in de up/down-status wordt door de tunnel geen dataverkeer doorgestuurd of verwerkt. Echter, het blijft het verzenden van keepalive pakketten. Op de ontvangst van een keepalive reactie, met de implicatie dat het tunneleindpunt opnieuw bereikbaar is, wordt de tunnelkeepalive teller teruggesteld aan 0, en het lijnprotocol over de tunnel komt omhoog.

Om keepalives in actie te zien, laat **debug tunnel** toe en **zuiver tunnelkeepalive**.

Steekproef debugt van router A:

```
debug tunnel keepalive
Tunnel keepalive debugging is on
01:19:16.019: Tunnel0: sending keepalive, 192.168.1.1->192.168.1.2
(len=24 ttl=0), counter=15
01:19:21.019: Tunnel0: sending keepalive, 192.168.1.1->192.168.1.2
(len=24 ttl=0), counter=16
01:19:26.019: Tunnel0: sending keepalive, 192.168.1.1->192.168.1.2
(len=24 ttl=0), counter=17
```

GRE-keepalives en Unicast-doorsturen van omgekeerde paden

Unicast RPF (Unicast Reverse Path Forwarding) is een beveiligingsfunctie die gespoofde IP-verkeer helpt detecteren en neerzetten met een validatie van het pakketbronadres tegen de routeringstabel. Wanneer Unicast RPF in strikte modus wordt uitgevoerd (**ip verifieer unicastbron reach-via rx**), moet het pakket worden ontvangen op de interface die de router zou gebruiken om het retourpakket door te sturen. Als strikte modus of losse modus Unicast RPF is ingeschakeld op de tunnelinterface van de router die de GRE keepalive-pakketten ontvangt, dan worden de keepalives-pakketten door RPF gedropt na tunneldecapsulation, omdat de route naar het bronadres van het pakket (router eigen tunnelbronadres) niet door de tunnelinterface loopt. De RPF-pakketdruppels kunnen als volgt worden waargenomen in de uitvoer van het **showip-verkeer**:

```
Router#show ip traffic | section Drop
Drop: 0 encapsulation failed, 0 unresolved, 0 no adjacency
0 no route, 156 unicast RPF, 0 forced drop
0 options denied
```

Als gevolg hiervan, de initiatiefnemer van de tunnel keepalives brengt de tunnel neer te wijten aan gemiste keepalives retourpakketten. Dus Unicast RPF moet niet worden geconfigureerd in strikte of losse modus zodat GRE tunnelkeepalives kunnen werken. Raadpleeg Unicast Reverse Path Forwarding voor meer informatie over Unicast RPF.

IPsec- en GRE-keepalives

GRE-tunnels met IPsec

GRE-tunnels worden soms gecombineerd met IPsec omdat IPsec IPsec IP-multicast pakketten niet ondersteunt. Daarom kunnen dynamische routeringsprotocollen niet succesvol via een IPsec VPN-netwerk worden uitgevoerd. Aangezien GRE-tunnels IP-multicast ondersteunen, kan een dynamisch routeringsprotocol worden uitgevoerd via een GRE-tunnel. De resulterende GRE IP-unicastpakketten kunnen worden versleuteld met IPsec.

Er zijn twee verschillende manieren waarop IPsec GRE-pakketten kan versleutelen:

- Een manier is met het gebruik van een cryptokaart. Wanneer een crypto-kaart wordt gebruikt, wordt deze toegepast op de uitgaande fysieke interface(s) voor de GRE-tunnelpakketten. In dit geval is de volgorde van de stappen als volgt:

  Versleuteld pakket bereikt de fysieke interface.Packet wordt gedecrypteerd en doorgestuurd naar de tunnelinterface.Packet wordt gedecapsuleerd en vervolgens doorgestuurd naar de IP-bestemming in duidelijke tekst.

- De andere manier is het gebruik van tunnelbescherming. Wanneer tunnelbescherming wordt gebruikt, wordt deze ingesteld op de GRE-tunnelinterface. De opdracht voor tunnelbescherming is beschikbaar gekomen in Cisco IOS-softwarerelease 12.2(13)T. In dit geval is de volgorde van de stappen als volgt:

  Versleuteld pakket bereikt fysieke interface.Packet wordt doorgestuurd naar de tunnelinterface.Packet wordt gedecrypteerd en gedecapsuleerd en vervolgens doorgestuurd naar de IP-bestemming in duidelijke tekst.

Beide methoden specificeren dat IPsec-codering wordt uitgevoerd na toevoeging van de GRE-insluiting. Er zijn twee belangrijke verschillen tussen wanneer u een cryptokaart gebruikt en wanneer u tunnelbescherming gebruikt:

- De IPsec crypto kaart is gekoppeld aan de fysieke interface en wordt gecontroleerd als pakketten worden doorgestuurd naar de fysieke interface.

  De GRE-tunnel heeft het pakket al tegen dit punt ingekapseld.
- Tunnelbeveiliging koppelt de coderingsfunctionaliteit aan de GRE-tunnel en wordt gecontroleerd nadat het pakket is ingesloten GRE, maar voordat het pakket aan de fysieke interface wordt geleverd.

Problemen met keepalives wanneer u IPsec en GRE combineert

Gezien de twee manieren om encryptie aan GRE-tunnels toe te voegen, zijn er drie verschillende manieren om een versleutelde GRE-tunnel in te stellen:

1. Peer A heeft tunnelbescherming die op de tunnelinterface wordt gevormd terwijl Peer B crypto kaart heeft die op de fysieke interface wordt gevormd.
2. Peer A heeft crypto kaart geconfigureerd op de fysieke interface terwijl Peer B tunnelbescherming heeft geconfigureerd op de tunnelinterface.
3. Beide peers hebben tunnelbescherming ingesteld op de tunnelinterface.

De configuratie beschreven in Scenarios 1 en 2 wordt vaak uitgevoerd in een hub-and-spoke ontwerp. Tunnelbeveiliging is ingesteld op de hubrouter om de omvang van de configuratie te beperken en op elke spaak wordt een statische cryptokaart gebruikt.

Overweeg elk van deze scenario's met GRE keepalives toegelaten op Peer B (spaak) en waar de tunnelmodus voor encryptie wordt gebruikt.

Scenario 1

Instelling:

- Peer A maakt gebruik van tunnelbescherming.
- Peer B maakt gebruik van Crypto Maps.
- Keepalives zijn ingeschakeld op peer B.
- IPsec-encryptie wordt uitgevoerd in tunnelmodus.

In dit scenario, aangezien GRE keepalives op Peer B worden gevormd, zijn de opeenvolgingsgebeurtenissen wanneer keepalive wordt geproduceerd als volgt:

1. Peer B genereert een keepalive-pakket dat is ingekapseld en vervolgens doorgestuurd naar de fysieke interface waar het wordt versleuteld en doorgestuurd naar de tunnelbestemming Peer A.

   Pakket verzonden van Peer B naar peer A:

2. Bij Peer A wordt de GRE keepalive ontcijferd:

   gedecapsuleerd:

   Vervolgens wordt het innerlijke GRE keepalive-responspakket gerouteerd op basis van het doeladres dat Peer B is. Dat betekent op Peer A, wordt het pakket onmiddellijk uit de fysieke interface aan Peer B. gerouteerd. Aangezien Peer A tunnelbescherming op de tunnelinterface gebruikt, wordt het keepalive pakket niet gecodeerd.

   Daarom wordt pakket verzonden van peer A naar peer B:

   Opmerking: keepalive is niet versleuteld.

3. Peer B ontvangt nu een GRE keepalive-respons die niet is versleuteld op zijn fysieke interface, maar vanwege de crypto-kaart die is geconfigureerd op de fysieke interface, verwacht het een versleuteld pakket en laat het dus vallen.

Daarom, alhoewel de Peer A op de wachtrijen antwoordt en router Peer B de reacties ontvangt, verwerkt het hen nooit en verandert uiteindelijk het lijnprotocol van de tunnelinterface in benedenstaat.

Resultaat:

Keepalives ingeschakeld op Peer B zorgt ervoor dat de tunnelstatus op Peer B verandert in omhoog/omlaag.

Scenario 2

Instelling:

- Peer A maakt gebruik van Crypto Maps.
- Peer B maakt gebruik van tunnelbescherming.
- Keepalives zijn ingeschakeld op peer B.
- IPsec-encryptie wordt uitgevoerd in tunnelmodus.

In dit scenario, aangezien GRE keepalives op Peer B worden gevormd, zijn de opeenvolgingsgebeurtenissen wanneer keepalive wordt geproduceerd als volgt:

1. Peer B genereert een keepalive-pakket dat is ingekapseld en vervolgens versleuteld door de tunnelbescherming op de tunnelinterface en vervolgens doorgestuurd naar de fysieke interface.

   Pakket verzonden van Peer B naar peer A:

2. Bij Peer A wordt de GRE keepalive ontcijferd:

   gedecapsuleerd:

   Vervolgens wordt het innerlijke GRE keepalive-responspakket gerouteerd op basis van het doeladres dat Peer B is. Dat betekent op Peer A, wordt het pakket onmiddellijk teruggeleid uit de fysieke interface aan Peer B. Aangezien Peer A crypto kaarten op de fysieke interface gebruikt, versleutelt het eerst dit pakket alvorens het het door:sturen.

   Daarom wordt pakket verzonden van peer A naar peer B:

   Opmerking: de keepalive-respons is versleuteld.

3. Peer B ontvangt nu een versleutelde GRE keepalive-respons waarvan de bestemming wordt doorgestuurd naar de tunnelinterface waar het wordt ontsleuteld:

   Aangezien Protocal Type op 0 is ingesteld, weet Peer B dat dit een keepalive-respons is en verwerkt het als zodanig.

Resultaat:

Keepalives ingeschakeld op Peer B bepalen met succes wat de tunnelstatus kan worden gebaseerd op de beschikbaarheid van de tunnelbestemming.

Scenario 3

Instelling:

- Beide peers gebruiken tunnelbescherming.
- Keepalives zijn ingeschakeld op peer B.
- IPsec-encryptie wordt uitgevoerd in tunnelmodus.

Dit scenario is vergelijkbaar met scenario 1 in die zin dat wanneer Peer A de versleutelde keepalive ontvangt, het decrypteert en decapsuleert het. Wanneer de reactie echter wordt doorgestuurd, wordt deze niet versleuteld omdat Peer A tunnelbescherming op de tunnelinterface

gebruikt. Aldus, laat Peer B de unencrypted keepalive reactie vallen en verwerkt het niet.

Resultaat:

Keepalives ingeschakeld op Peer B zorgt ervoor dat de tunnelstatus op Peer B verandert in omhoog/omlaag.

Tijdelijke oplossing

In dergelijke situaties waarin de GRE-pakketten moeten worden versleuteld, zijn er drie mogelijke oplossingen:

1. Gebruik een crypto kaart op Peer A, tunnelbescherming op Peer B, en laat keepalives op Peer B toe.

   Aangezien dit type van configuratie meestal wordt gebruikt in hub-and-spoke opstellingen en omdat in dergelijke opstellingen het belangrijker is voor de spoke om zich bewust te zijn van de hubs bereikbaarheid, is de oplossing om een dynamische crypto kaart te gebruiken op de hub (Peer A) en tunnelbescherming op de spoke (Peer B) en GRE te laten behouden op de spoke. Op deze manier, hoewel de GRE-tunnelinterface op de hub omhoog blijft, zijn de routerbuur en de routes door de tunnel verloren en kan de alternatieve route worden vastgesteld. Op de spaak, het feit dat de tunnelinterface daalde kan het teweegbrengen om een dialer interface omhoog te brengen en terug te bellen naar de hub (of een andere router bij de hub), dan een nieuwe verbinding te vestigen.

2. Gebruik iets anders dan GRE keepalives om peer bereikbaarheid te bepalen.

   Als beide routers met tunnelbescherming worden geconfigureerd, kunnen GRE-tunnelmeubelen niet in beide richtingen worden gebruikt. In dit geval is de enige optie om het routeringsprotocol of ander mechanisme, zoals de Service Assurance Agent, te gebruiken om te ontdekken of de peer bereikbaar is of niet.

3. Gebruik crypto maps op peer A en peer B.

   Als beide routers zijn geconfigureerd met cryptokaarten, kunnen de tunnelkeepalives doorheen in beide richtingen en de GRE-tunnelinterfaces kunnen in beide of beide richtingen worden gesloten en een reserveverbinding tot stand brengen. Dit is de meest flexibele optie.

Gerelateerde informatie

- RFC 1701, Generic Router Encapsulation (GRE)
- RFC 2890: Uitbreidingen van sleutel- en volgnummer naar GRE
- Generic Routing Encapsulation (GRE)-tunnelbehoud
- IP-fragmentatie en PMTUD
- Overzicht van Keepalive-mechanismen op Cisco IOS
- Technische ondersteuning – Cisco Systems

Over deze vertaling

Cisco heeft dit document vertaald via een combinatie van machine- en menselijke technologie om onze gebruikers wereldwijd ondersteuningscontent te bieden in hun eigen taal. Houd er rekening mee dat zelfs de beste machinevertaling niet net zo nauwkeurig is als die van een professionele vertaler. Cisco Systems, Inc. is niet aansprakelijk voor de nauwkeurigheid van deze vertalingen en raadt aan altijd het oorspronkelijke Engelstalige document (link) te raadplegen.