GRE 터널 킵얼라이브 이해

목차

소개

이 문서에서는 GRE(Generic Routing Encapsulation) 킵얼라이브의 정의 및 작동 방식에 대해 설명합니다.

GRE 터널

GRE 터널은 전송 프로토콜 내에서 승객 패킷을 캡슐화하는 방법을 제공하는 Cisco 라우터의 논리적 인터페이스입니다. 포인트-투-포인트(point-to-point) 캡슐화 방식을 구현하기 위해 서비스를 제공하도록 설계된 아키텍처입니다.

GRE 터널은 완전히 스테이트리스(stateless)로 설계되었습니다. 즉, 각 터널 엔드포인트가 원격 터널 엔드포인트의 상태 또는 가용성에 대한 정보를 유지하지 않습니다. 따라서 터널의 원격 끝에 연결할 수 없는 경우 로컬 터널 엔드포인트 라우터가 GRE 터널 인터페이스의 라인 프로토콜을 작동 중지시킬 수 없게 됩니다. 링크의 원격 끝을 사용할 수 없을 때 인터페이스를 중단으로 표시하는 기능은 해당 인터페이스를 아웃바운드 인터페이스로 사용하는 라우팅 테이블에서 경로(특히 고정 경로)를 제거하기 위해 사용됩니다. 특히 인터페이스에 대한 라인 프로토콜이 down으로 변경되면 해당 인터페이스를 가리키는 모든 고정 경로가 라우팅 테이블에서 제거됩니다. 이렇게 하면 대체(부동) 고정 경로를 설치하거나 대체 다음 홉 또는 인터페이스를 선택하기 위해 PBR(Policy Based Routing)을 설치할 수 있습니다.

일반적으로 GRE 터널 인터페이스는 구성되는 즉시 나타나며, 유효한 터널 소스 주소 또는 인터페이스가 가동 상태인 경우 그대로 유지됩니다. 터널 대상 IP 주소도 라우팅 가능해야 합니다. 이는 터널의 다른 쪽이 구성되지 않은 경우에도 마찬가지입니다. 즉, GRE 터널 패킷이 터널의 다른 쪽 끝에 도달하지 않더라도 GRE 터널 인터페이스를 통한 패킷의 고정 경로 또는 PBR 전달은 계속 유효합니다.

GRE keepalive가 구현되기 전에는 라우터의 로컬 문제를 확인할 수 있는 방법만 있었고 중간 네트워크의 문제를 확인할 수 있는 방법은 없었습니다. 예를 들어, GRE 터널링 패킷이 성공적으로 전달

되지만 터널의 다른 끝에 도달하기 전에 손실되는 경우가 있습니다. 이러한 시나리오에서는 PBR을 사용하는 대체 경로 또는 다른 인터페이스를 통한 유동 고정 경로를 사용할 수 있더라도 GRE 터널을 통과하는 데이터 패킷이 "블랙홀"이 됩니다. GRE 터널 인터페이스의 Keepalive는 물리적 인터페이스에서 Keepalive를 사용하는 것과 동일한 방법으로 이 문제를 해결하기 위해 사용됩니다.

> **참고**: GRE 킵얼라이브는 어떤 상황에서도 IPsec 터널 보호와 함께 지원되지 않습니다. 이 문서에서는 이 문제에 대해 설명합니다.

터널 킵얼라이브의 작동 방식

GRE 터널 킵얼라이브 메커니즘은 원격 라우터가 GRE 킵얼라이브를 지원하지 않는 경우에도 원격 라우터에서 킵얼라이브 패킷을 수신하고 시작할 수 있는 기능을 한쪽에서 제공한다는 점에서 PPP 킵얼라이브와 유사합니다. GRE는 IP 내부 IP를 터널링하는 패킷 터널링 메커니즘이므로 GRE IP 터널 패킷을 다른 GRE IP 터널 패킷 내부에 구축할 수 있습니다. GRE keepalive의 경우 발신자는 원래 keepalive 요청 패킷 내부에 keepalive 응답 패킷을 미리 구축하므로 원격 단은 외부 GRE IP 헤더의 표준 GRE 역캡슐화만 수행한 다음 내부 IP GRE 패킷을 발신자에게 되돌립니다. 이러한 패킷은 IP 터널링 개념을 보여줍니다. 여기서 GRE는 캡슐화 프로토콜이고 IP는 전송 프로토콜입니다. 승객 프로토콜도 IP입니다(Decnet, IPX(Internetwork Packet Exchange) 또는 Appletalk와 같은 다른 프로토콜일 수 있음).

일반 패킷:

IP 헤더	TCP 헤더	Telnet

터널링 패킷:

GRE IP 헤더	GRE	IP 헤더	TCP 헤더	Telnet

- IP는 전송 프로토콜입니다.
- GRE는 캡슐화 프로토콜입니다.
- IP는 승객 프로토콜입니다.

다음은 라우터 A에서 시작되어 라우터 B를 대상으로 하는 킵얼라이브 패킷의 예입니다. 라우터 B가 라우터 A에 반환하는 keepalive 응답이 이미 내부 IP 헤더 내에 있습니다. 라우터 B는 단순히 keepalive 패킷을 역캡슐화하여 물리적 인터페이스로 다시 보냅니다(S2). 다른 GRE IP 데이터 패킷과 마찬가지로 GRE keepalive 패킷을 처리합니다.

GRE 킵얼라이브:

GRE IP 헤더		GRE		IP 헤더		GRE
소스 A	대상 B	PT=IP	소스 B	대상 A		PT=0

이 메커니즘은 keepalive 응답이 터널 인터페이스가 아닌 물리적 인터페이스를 포워드아웃하도록 합니다. 즉, GRE 킵얼라이브 응답 패킷은 '터널 보호 ...', QoS, VRF(Virtual Routing and Forwarding) 등과 같은 터널 인터페이스의 출력 기능의 영향을 받지 않습니다.

> **참고**: GRE 터널 인터페이스의 인바운드 ACL(Access Control List)이 구성된 경우 반대 디바이스에서 전송하는 GRE 터널 킵얼라이브 패킷이 허용되어야 합니다. 그렇지 않으면 반대편 디바이스 GRE 터널이 다운됩니다. (**access-list** <number> **permit gre host** <tunnel-source> **host** <tunnel-destination>)

GRE 터널 킵얼라이브의 또 다른 특성은 각 측의 킵얼라이브 타이머가 독립적이며 일치하지 않아도 된다는 것이며, 이는 PPP 킵얼라이브와 비슷합니다.

> **팁**: 터널의 한쪽에서만 keepalive를 구성하는 데 문제가 있는 것은 keepalive 타이머가 만료될 경우 keepalive가 구성된 라우터만 터널 인터페이스가 다운된 것으로 표시된다는 것입니다. keepalive가 구성되지 않은 다른 쪽의 GRE 터널 인터페이스는 터널의 다른 쪽이 다운되더라도 계속 업상태를 유지합니다. 터널은 킵얼라이브가 구성되지 않은 측면에서 터널로 전달되는 패킷의 블랙홀이 될 수 있습니다.

> **팁**: 대규모 허브-스포크 GRE 터널 네트워크에서는 허브 측이 아닌 스포크 측에서만 GRE 킵얼라이브를 구성하는 것이 적절할 수 있습니다. 이는 종종 스포크가 허브에 연결할 수 없으므로 백업 경로(예: 다이얼 백업)로 전환하는 것이 더 중요하기 때문입니다.

GRE 터널 킵얼라이브

Cisco IOS® Software Release 12.2(8)T를 사용하면 포인트-투-포인트 GRE 터널 인터페이스에 keepalive를 구성할 수 있습니다. 이러한 변경으로 인해 일정 기간 동안 keepalive에 장애가 발생하면 터널 인터페이스가 동적으로 종료됩니다.

다른 형태의 keepalive 작동 방식에 대한 자세한 내용은 Cisco IOS의 Keepalive 메커니즘 개요를 참조하십시오.

> **참고**: GRE 터널 킵얼라이브는 포인트-투-포인트 GRE 터널에서만 지원됩니다. mGRE(multipoint GRE) 터널에서 터널 킵얼라이브를 구성할 수 있지만 효과가 없습니다.

> **참고**: 일반적으로 터널 인터페이스 및 fVRF에서 VRF가 사용되는 경우 터널 킵얼라이브가 작동하지 않습니다('tunnel vrf ...') 및 iVRF('ip vrf forwarding ...'(터널 인터페이스의 경우) 일치하지 않습니다. 이는 keepalive를 다시 요청자에게 "반영"하는 터널 엔드포인트에서 중요합니다. keepalive 요청이 수신되면 fVRF에서 수신되어 캡슐화 해제됩니다. 이렇게 하면 미리 만들어진 keepalive 회신이 드러나며, 이 회신은 발신자에게 다시 전달되어야 하지만 해당 전달은 터널 인터페이스의 iVRF 컨텍스트에 있습니다. 따라서 iVRF와 fVRF가 일치하지 않으면 keepalive 응답 패킷이 발신자에게 다시 전달되지 않습니다. 이는 iVRF 및/또는 fVRF를 "global"로 교체한 경우에도 마찬가지입니다.

이 출력은 GRE 터널에서 keepalive를 구성하기 위해 사용하는 명령을 보여줍니다.

```
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.
```

터널 킵얼라이브 메커니즘의 작동 방식을 자세히 알아보려면 다음 터널 토폴로지 및 컨피그레이션

예를 고려하십시오.

라우터 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
```

라우터 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
```

이 시나리오에서 라우터 A는 다음 단계를 수행합니다.

1. 다음과 같은 경우 5초마다 내부 IP 헤더를 구성합니다.

   소스가 로컬 터널 대상(192.168.1.2)으로 설정됩니다대상은 192.168.1.1인 로컬 터널 소스로 설정됩니다.

   GRE 헤더가 PT(Protocol Type)가 0인 상태로 추가됩니다

   라우터 A에서 생성되었지만 전송되지 않은 패킷:

2. 해당 패킷을 터널 인터페이스 외부로 전송합니다. 그러면 외부 IP 헤더로 패킷이 캡슐화됩니다.

   소스는 192.168.1.1인 로컬 터널 소스로 설정됩니다.대상은 로컬 터널 대상(192.168.1.2)으로 설정됩니다

GRE 헤더가 PT = IP와 함께 추가됩니다.

라우터 A에서 라우터 B로 전송된 패킷:

3. 터널 킵얼라이브 카운터를 1씩 늘립니다.

4. 종단 터널 엔드포인트에 도달할 수 있는 방법이 있으며 다른 이유로 인해 터널 회선 프로토콜이 중단되지 않았다고 가정할 경우 패킷이 라우터 B에 도착합니다. 그런 다음 터널 0과 일치하고, 캡슐화 해제되어 라우터 A의 터널 소스 IP 주소인 대상 IP로 전달됩니다.

   라우터 B에서 라우터 A로 전송:

5. 라우터 A에 도착하면 패킷이 역캡슐화되고 PT를 확인하면 0이 됩니다. 이는 keepalive 패킷임을 나타냅니다. 그런 다음 터널 킵얼라이브 카운터가 0으로 재설정되고 패킷이 삭제됩니다.

라우터 B에 연결할 수 없는 경우, 라우터 A는 일반 트래픽은 물론 keepalive 패킷을 계속 구성하고 전송합니다. keepalive가 다시 돌아오지 않으면 터널 킵얼라이브 카운터가 재시도 횟수보다 작은 한 터널 회선 프로토콜은 계속 작동합니다. 이 경우에는 4회입니다. 이 조건이 참이 아닌 경우, 다음에 라우터 A가 라우터 B에 킵얼라이브를 보내려고 시도할 때 회선 프로토콜이 중단됩니다.

참고: up/down 상태에서는 터널이 데이터 트래픽을 전달하거나 처리하지 않습니다. 그러나 keepalive 패킷은 계속 전송합니다. keepalive 응답을 수신하면 터널 엔드포인트에 다시 연결할 수 있다는 의미와 함께 터널 keepalive 카운터가 0으로 재설정되고 터널의 회선 프로토콜이 시작됩니다.

keepalive의 작동 상태를 확인하려면 debug tunnel 및 **debug tunnel keepalive**를 **활성화합니다**.

라우터 A의 샘플 디버깅:

```
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 킵얼라이브 및 유니캐스트 역방향 경로 전달

유니캐스트 RPF(Unicast Reverse Path Forwarding)는 라우팅 테이블을 기준으로 패킷 소스 주소를 검증하여 스푸핑된 IP 트래픽을 탐지하고 삭제하는 데 도움이 되는 보안 기능입니다. 유니캐스트 RPF가 엄격한 모드(**ip verify unicast source reachable-via rx**)에서 실행되는 경우, 반환 패킷을 전달하기 위해 라우터가 사용할 인터페이스에서 패킷을 수신해야 합니다. 엄격한 모드 또는 느슨한 모드 유니캐스트 RPF가 GRE 킵얼라이브 패킷을 수신하는 라우터의 터널 인터페이스에서 활성화된 경우 패킷의 소스 주소(라우터 자체 터널 소스 주소)에 대한 경로가 터널 인터페이스를 통하지 않기 때문에 터널 역캡슐화 후 킵얼라이브 패킷이 RPF에 의해 삭제됩니다. RPF 패킷 삭제는 다음과 같이 **show ip traffic 출력**에서 관찰될 수 있습니다.

```
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
```
따라서 누락된 킵얼라이브 반환 패킷으로 인해 터널 킵얼라이브의 개시자가 터널을 종료합니다. 따라서 GRE 터널 킵얼라이브가 작동하려면 유니캐스트 RPF를 엄격한 모드 또는 느슨한 모드로 구성하지 않아야 합니다. Unicast RPF에 대한 자세한 내용은 Unicast Reverse Path Forwarding 이해를 참조하십시오.

IPsec 및 GRE 킵얼라이브

IPsec을 사용하는 GRE 터널

IPsec에서 IP 멀티캐스트 패킷을 지원하지 않기 때문에 GRE 터널이 IPsec과 결합되는 경우가 있습니다. 이로 인해 동적 라우팅 프로토콜은 IPsec VPN 네트워크를 통해 성공적으로 실행될 수 없습니다. GRE 터널은 IP 멀티캐스트를 지원하므로 GRE 터널을 통해 동적 라우팅 프로토콜을 실행할 수 있습니다. 결과로 생성되는 GRE IP 유니캐스트 패킷은 IPsec에 의해 암호화될 수 있습니다.

IPsec에서 GRE 패킷을 암호화하는 방법에는 두 가지가 있습니다.

- 한 가지 방법은 암호화 맵을 사용하는 것입니다. 암호화 맵이 사용될 경우 GRE 터널 패킷의 아웃바운드 물리적 인터페이스에 적용됩니다. 이 경우 단계의 순서는 다음과 같습니다.

  암호화된 패킷이 물리적 인터페이스에 도달합니다.패킷이 해독되고 터널 인터페이스에 전달됩니다.패킷은 역캡슐화된 다음 일반 텍스트로 IP 대상에 전달됩니다.

- 다른 방법은 터널 보호를 사용하는 것입니다. 터널 보호를 사용하는 경우 GRE 터널 인터페이스에 구성됩니다. tunnel protection 명령은 Cisco IOS Software Release 12.2(13)T에서 사용할 수 있게 되었습니다. 이 경우 단계의 순서는 다음과 같습니다.

  암호화된 패킷이 물리적 인터페이스에 도달합니다.패킷은 터널 인터페이스로 전달됩니다.패킷이 암호 해독되고 역캡슐화된 다음 일반 텍스트로 IP 대상에 전달됩니다.

두 방법 모두 GRE 캡슐화를 추가한 후에 IPsec 암호화를 수행하도록 지정합니다. 암호화 맵을 사용하는 경우와 터널 보호를 사용하는 경우 사이에는 두 가지 중요한 차이점이 있습니다.

- IPsec 암호화 맵은 물리적 인터페이스에 연결되며, 패킷이 물리적 인터페이스 외부로 전달될 때 확인됩니다.

  GRE 터널은 이 시점에 패킷을 이미 GRE 캡슐화했습니다.
- 터널 보호는 암호화 기능을 GRE 터널에 연결하고, 패킷이 GRE 캡슐화된 후 패킷이 물리적 인터페이스로 전달되기 전에 확인됩니다.

IPsec과 GRE를 결합할 때 Keepalive에 문제가 있음

GRE 터널에 암호화를 추가하는 두 가지 방법을 고려할 때, 세 가지 방법으로 암호화된 GRE 터널을 설정할 수 있습니다.

1. 피어 A는 터널 인터페이스에 터널 보호가 구성되어 있는 반면 피어 B는 물리적 인터페이스에

암호화 맵이 구성되어 있습니다.
2. 피어 A는 물리적 인터페이스에 암호화 맵이 구성되어 있고 피어 B는 터널 인터페이스에 터널 보호가 구성되어 있습니다.
3. 두 피어 모두 터널 인터페이스에 터널 보호가 구성되어 있습니다.

시나리오 1과 2에 설명된 컨피그레이션은 허브 앤 스포크 설계로 수행되는 경우가 많습니다. 컨피그레이션의 크기를 줄이기 위해 허브 라우터에 터널 보호가 구성되고 각 스포크에 고정 암호화 맵이 사용됩니다.

피어 B(스포크)에서 GRE 킵얼라이브가 활성화되고 터널 모드가 암호화에 사용되는 각각의 시나리오를 고려해 보십시오.

시나리오 1

설정:

- 피어 A는 터널 보호를 사용합니다.
- 피어 B는 암호화 맵을 사용합니다.
- Keepalive는 피어 B에서 활성화됩니다.
- IPsec 암호화는 터널 모드에서 수행됩니다.

이 시나리오에서는 GRE 킵얼라이브가 피어 B에서 구성되므로 킵얼라이브가 생성될 때 발생하는 시퀀스 이벤트는 다음과 같습니다.

1. 피어 B는 GRE가 캡슐화된 다음 물리적 인터페이스에 전달되는 킵얼라이브 패킷을 생성하며, 여기서 암호화되어 터널 대상 피어 A로 전송됩니다.

   피어 B에서 피어 A로 전송된 패킷:

2. 피어 A에서 GRE 킵얼라이브가 해독됩니다.

   캡슐화 해제:

   그러면 내부 GRE 킵얼라이브 응답 패킷이 대상 주소인 피어 B를 기반으로 라우팅됩니다. 즉, 피어 A에서는 패킷이 물리적 인터페이스에서 피어 B로 즉시 다시 라우팅됩니다. 피어 A는 터널 인터페이스에서 터널 보호를 사용하므로 keepalive 패킷은 암호화되지 않습니다.

   따라서 피어 A에서 피어 B로 전송되는 패킷은 다음과 같습니다.

   참고: keepalive는 암호화되지 않습니다.

3. 이제 피어 B는 물리적 인터페이스에서 암호화되지 않은 GRE keepalive 응답을 수신하지만, 물리적 인터페이스에 구성된 암호화 맵 때문에 암호화된 패킷을 예상하므로 삭제합니다.

따라서 피어 A가 키 인터페이스에 응답하고 라우터 피어 B가 응답을 수신하더라도 이를 처리하지

않고 결국 터널 인터페이스의 라인 프로토콜을 중단 상태로 변경합니다.

결과:

피어 B에서 킵얼라이브를 활성화하면 피어 B의 터널 상태가 up/down으로 변경됩니다.

시나리오 2

설정:

- 피어 A는 암호화 맵을 사용합니다.
- 피어 B는 터널 보호를 사용합니다.
- Keepalive는 피어 B에서 활성화됩니다.
- IPsec 암호화는 터널 모드에서 수행됩니다.

이 시나리오에서는 GRE 킵얼라이브가 피어 B에 구성되므로 킵얼라이브가 생성될 때 발생하는 시퀀스 이벤트는 다음과 같습니다.

1. 피어 B는 GRE가 캡슐화된 다음 터널 인터페이스의 터널 보호에 의해 암호화된 다음 물리적 인터페이스로 전달되는 킵얼라이브 패킷을 생성합니다.

   피어 B에서 피어 A로 전송된 패킷:

2. 피어 A에서 GRE 킵얼라이브가 해독됩니다.

   캡슐화 해제:

   그러면 내부 GRE 킵얼라이브 응답 패킷이 대상 주소인 피어 B를 기반으로 라우팅됩니다. 즉, 피어 A에서는 패킷이 물리적 인터페이스에서 피어 B로 즉시 다시 라우팅됩니다. 피어 A는 물리적 인터페이스에서 암호화 맵을 사용하므로 패킷을 전달하기 전에 먼저 이 패킷을 암호화합니다.

   따라서 피어 A에서 피어 B로 전송되는 패킷은 다음과 같습니다.

   참고: keepalive 응답은 암호화됩니다.

3. 피어 B는 이제 목적지가 해독되는 터널 인터페이스로 전달되는 암호화된 GRE 킵얼라이브 응답을 수신합니다.

   Protocal Type(프로토콜 유형)이 0으로 설정되어 있으므로 피어 B는 이 응답이 킵얼라이브 응답임을 알고 이를 처리합니다.

결과:

피어 B에서 활성화된 keepalives는 터널 대상의 가용성을 기반으로 어떤 터널 상태를 설정할 수 있는지 성공적으로 결정합니다.

시나리오 3

설정:

- 두 피어 모두 터널 보호를 사용합니다.
- Keepalive는 피어 B에서 활성화됩니다.
- IPsec 암호화는 터널 모드에서 수행됩니다.

이 시나리오는 피어 A가 암호화된 keepalive를 수신하면 이를 해독하고 역캡슐화한다는 점에서 시나리오 1과 유사합니다. 그러나 응답이 다시 전달되면 피어 A가 터널 인터페이스에서 터널 보호를 사용하므로 암호화되지 않습니다. 따라서 피어 B는 암호화되지 않은 keepalive 응답을 삭제하고 처리하지 않습니다.

결과:

피어 B에서 킵얼라이브를 활성화하면 피어 B의 터널 상태가 up/down으로 변경됩니다.

해결 방법

GRE 패킷을 암호화해야 하는 상황에서는 다음과 같은 세 가지 해결 방법이 있습니다.

1. 피어 A에서 암호화 맵을 사용하고, 피어 B에서 터널 보호를 수행하고, 피어 B에서 keepalive를 활성화합니다.

   이 컨피그레이션 유형은 허브-앤-스포크 설정에서 주로 사용되고, 이러한 설정에서는 스포크가 허브 연결성을 인식하는 것이 더 중요하므로, 솔루션은 허브(피어 A)에서 동적 암호화 맵을 사용하고 스포크(피어 B)에서 터널 보호를 수행하고 스포크에서 GRE keepalive를 활성화하는 것입니다. 이렇게 하면 허브의 GRE 터널 인터페이스가 가동 상태로 유지되지만 라우팅 네이버와 터널을 통과하는 경로가 손실되고 대체 경로를 설정할 수 있습니다. 스포크에서 터널 인터페이스가 다운되었다는 사실은 다이얼러 인터페이스를 시작하고 허브(또는 허브의 다른 라우터)에 다시 전화를 건 다음 새 연결을 설정하도록 트리거할 수 있습니다.

2. 피어 연결 가능성을 확인하기 위해 GRE 킵얼라이브 이외의 다른 방법을 사용합니다.

   두 라우터가 모두 터널 보호로 구성된 경우 GRE 터널 킵얼라이브는 어느 방향으로도 사용할 수 없습니다. 이 경우 유일한 옵션은 피어에 연결할 수 있는지 여부를 확인하기 위해 라우팅 프로토콜 또는 기타 메커니즘(예: Service Assurance Agent)을 사용하는 것입니다.

3. 피어 A 및 피어 B에서 암호화 맵을 사용합니다.

두 라우터가 모두 암호화 맵으로 구성된 경우, 터널 킵얼라이브는 양방향으로 통과할 수 있으며 GRE 터널 인터페이스는 어느 한 방향이나 양방향으로 종료되어 백업 연결을 생성할 수 있습니다. 이것이 가장 유연한 옵션입니다.

관련 정보

- RFC 1701, GRE(Generic Router Encapsulation)
- RFC 2890, GRE에 대한 키 및 시퀀스 번호 확장
- GRE(Generic Routing Encapsulation) 터널 킵얼라이브
- IP 프래그먼트화 및 PMTUD
- Cisco IOS의 Keepalive 메커니즘 개요
- Technical Support - Cisco Systems

이 번역에 관하여

Cisco는 전 세계 사용자에게 다양한 언어로 지원 콘텐츠를 제공하기 위해 기계 번역 기술과 수작업 번역을 병행하여 이 문서를 번역했습니다. 아무리 품질이 높은 기계 번역이라도 전문 번역가의 번역 결과물만큼 정확하지는 않습니다. Cisco Systems, Inc.는 이 같은 번역에 대해 어떠한 책임도 지지 않으며 항상 원본 영문 문서(링크 제공됨)를 참조할 것을 권장합니다.