Beispielkonfigurationen und Debuggen für IPSec over Cable

Inhalt

Einführung

Internet Protocol Security (IPsec) ist ein Framework offener Standards, das die sichere private Kommunikation über IP-Netzwerke sicherstellt. Auf der Grundlage von Standards, die von der Internet Engineering Task Force (IETF) entwickelt wurden, gewährleistet IPsec Vertraulichkeit, Integrität und Authentizität der Datenkommunikation in einem öffentlichen IP-Netzwerk. IPsec stellt eine notwendige Komponente für eine standardbasierte, flexible Lösung zur Bereitstellung einer netzwerkweiten Sicherheitsrichtlinie dar.

Dieses Dokument enthält ein Konfigurationsbeispiel für IPsec zwischen zwei Cisco Kabelmodems. Bei dieser Konfiguration wird ein Verschlüsselungstunnel in einem Kabelnetzwerk zwischen zwei Cisco uBR9xx-Kabelmodemroutern erstellt. Der gesamte Datenverkehr zwischen den beiden Netzwerken wird verschlüsselt. Datenverkehr, der für andere Netzwerke bestimmt ist, darf jedoch unverschlüsselt weitergeleitet werden. Für Benutzer in kleinen Büros und Heimbüros (SOHO) ist damit die Erstellung von VPNs (Virtual Private Networks) über ein Kabelnetzwerk möglich.

Voraussetzungen

Anforderungen

Für dieses Dokument bestehen keine speziellen Anforderungen.

Verwendete Komponenten

Die Modems müssen diesen Anforderungen entsprechen, um IPsec auf zwei Kabelmodems zu konfigurieren:

- Cisco uBR904, uBR905 oder uBR924 im Routing-Modus
- IPsec 56-Feature-Set
- Cisco IOS® Softwareversion 12.0(5)T oder höher

Darüber hinaus müssen Sie über ein Cable Modem Termination System (CMTS) verfügen, d. h. einen DOCSIS-kompatiblen Headend-Kabelrouter (Data-over-Cable Service Interface Specifications), z. B. Cisco uBR7246, Cisco uBR7223 oder Cisco uBR7246VXR.

Die Informationen in diesem Dokument wurden von den Geräten in einer bestimmten Laborumgebung erstellt. Alle in diesem Dokument verwendeten Geräte haben mit einer leeren (Standard-)Konfiguration begonnen. Wenn Ihr Netzwerk in Betrieb ist, stellen Sie sicher, dass Sie die potenziellen Auswirkungen eines Befehls verstehen.

Konventionen

Weitere Informationen zu Dokumentkonventionen finden Sie unter Cisco Technical Tips Conventions (Technische Tipps zu Konventionen von Cisco).

Hintergrundtheorie

Im Beispiel in diesem Dokument werden ein uBR904-Kabelmodem, ein uBR924-Kabelmodem und ein uBR7246VXR CMTS verwendet. Auf den Kabelmodems wird die Cisco IOS Software Version 12.1(6) ausgeführt, und auf dem CMTS wird die Cisco IOS Software Release 12.1(4)EC ausgeführt.

Hinweis: In diesem Beispiel erfolgt die manuelle Konfiguration der Kabelmodems über den Konsolenport. Wenn ein automatisierter Prozess über die DOCSIS-Konfigurationsdatei durchgeführt wird (das Skript ios.cfg wird mit der IPsec-Konfiguration erstellt), *können* die Zugriffslisten 100 und 101 nicht verwendet werden. Dies liegt daran, dass die Cisco Implementierung des Simple Network Management Protocol (SNMP) docsDevNmAccess-Tabellen Cisco IOS-Zugriffslisten verwendet. Pro Schnittstelle wird eine Zugriffsliste erstellt. Auf uBR904, 924 und 905 werden die ersten beiden Zugriffslisten im Allgemeinen verwendet (100 und 101). Auf einem Kabelmodem, das USB (Universal Serial Bus) unterstützt, wie dem CVA120, werden drei Zugriffslisten verwendet (100, 101 und 102).

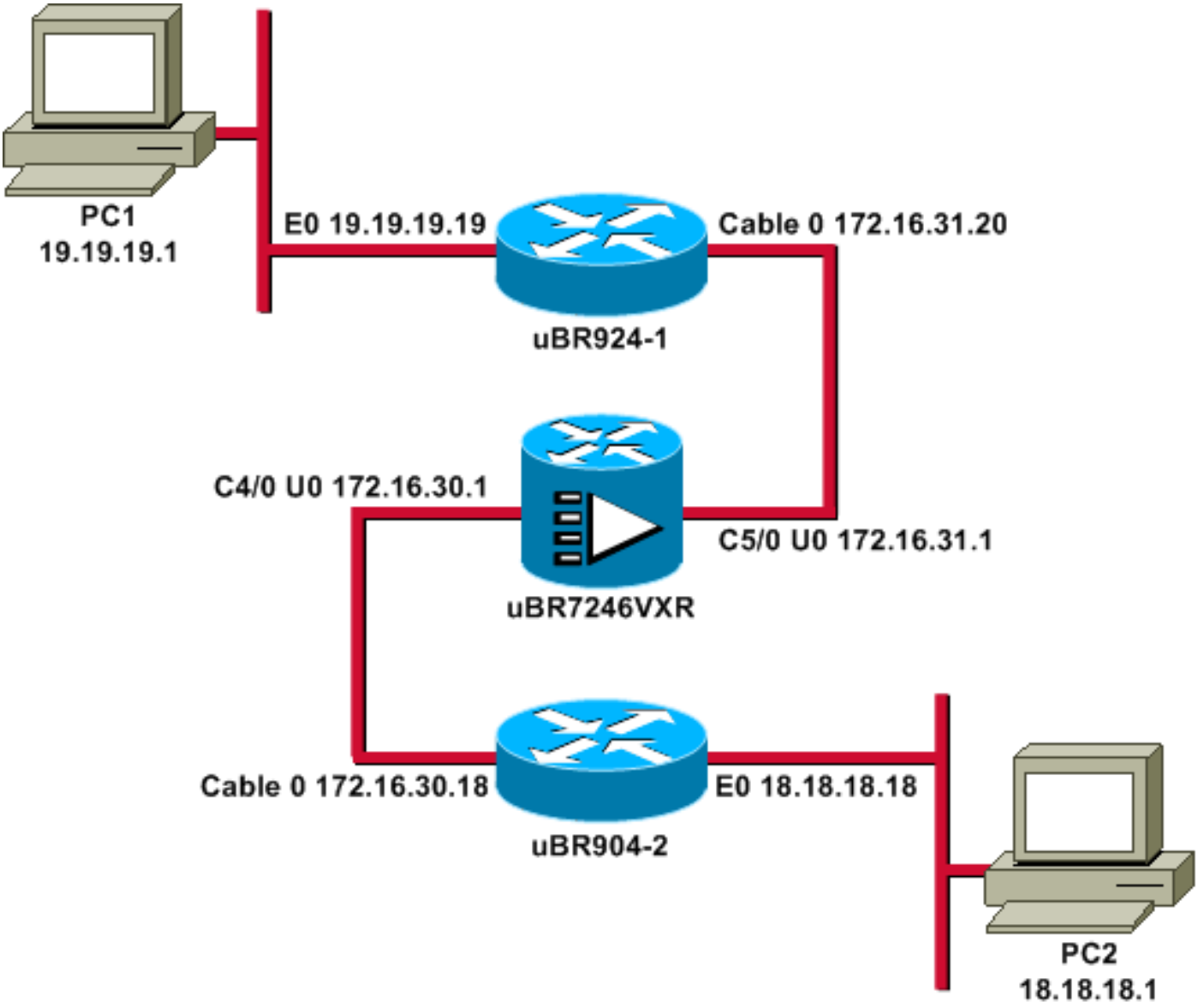
Konfigurieren

In diesem Abschnitt erhalten Sie Informationen zum Konfigurieren der in diesem Dokument beschriebenen Funktionen.

Hinweis: Verwenden Sie das Command Lookup Tool (nur registrierte Kunden), um weitere Informationen zu den Befehlen in diesem Dokument zu erhalten.

Netzwerkdiagramm

In diesem Dokument wird die folgende Netzwerkeinrichtung verwendet:



Hinweis: Alle IP-Adressen in diesem Diagramm haben eine 24-Bit-Maske.

Konfigurationen

In diesem Dokument werden folgende Konfigurationen verwendet:

- uBR924-1
- uBR904-2
- uBR7246VXR

uBR924-1

```
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname ubr924-1
!
enable password ww
!
!
!
```

```
!
clock timezone - -8
ip subnet-zero
no ip finger
!
ip audit notify log
ip audit po max-events 100
!
!
crypto isakmp policy 10
!--- Creates an Internet Key Exchange (IKE) policy with
the specified priority !--- number of 10. The range for
the priority is 1 to 10000, where 1 is the !--- highest
priority. This command also enters Internet Security
Association !--- and Key Management Protocol (ISAKMP)
policy configuration command mode. hash md5
!--- Specifies the MD5 (HMAC variant) hash algorithm for
packet authentication. authentication pre-share
!--- Specifies that the authentication keys are pre-
shared, as opposed to !--- dynamically negotiated using
Rivest, Shamir, and Adelman (RSA) public !--- key
signatures. group 2
!--- Diffie-Hellman group for key negotiation. lifetime
3600
!--- Defines how long, in seconds, each security
association should exist before !--- it expires. Its
range is 60 to 86400, and in this case, it is 1 hour.
crypto isakmp key mykey address 18.18.18.18
!--- Specifies the pre-shared key that should be used
with the peer at the !--- specific IP address. The key
can be any arbitrary alphanumeric key up to !--- 128
characters. The key is case-sensitive and must be
entered identically !--- on both routers. In this case,
the key is mykey and the peer is the !--- Ethernet
address of uBR904-2
.
!
crypto IPSec transform-set TUNNELSET ah-md5-hmac esp-des
!--- Establishes the transform set to use for IPsec
encryption. As many as !--- three transformations can be
specified for a set. Authentication Header !--- and ESP
are in use. Another common transform set used in
industry is !--- esp-des esp-md5-hmac.

!
crypto map MYMAP local-address Ethernet0
!--- Creates the MYMAP crypto map and applies it to the
Ethernet0 interface.

crypto map MYMAP 10 ipsec-isakmp
!--- Creates a crypto map numbered 10 and enters crypto
map configuration mode. set peer 18.18.18.18
!--- Identifies the IP address for the destination peer
router. In this case, !--- the Ethernet interface of the
remote cable modem (ubr904-2) is used. set transform-set
TUNNELSET
!--- Sets the crypto map to use the transform set
previously created. match address 101
!--- Sets the crypto map to use the access list that
specifies the type of !--- traffic to be encrypted. !---
Do not use access lists 100, 101, and 102 if the IPsec
config is !--- downloaded through the ios.cfg in the
DOCSIS configuration file.
```

```
!
!
!
!
voice-port 0
 input gain -2
 output attenuation 0
!
voice-port 1
 input gain -2
 output attenuation 0
!
!
!
interface Ethernet0
 ip address 19.19.19.19 255.255.255.0
 ip rip send version 2
 ip rip receive version 2
 no ip route-cache
 no ip mroute-cache
!
interface cable-modem0
 ip rip send version 2
 ip rip receive version 2
 no ip route-cache
 no ip mroute-cache
 cable-modem downstream saved channel 525000000 39 1
 cable-modem mac-timer t2 40000
 no cable-modem compliant bridge
 crypto map MYMAP
!--- Applies the previously created crypto map to the
cable interface. ! router rip version 2 network 19.0.0.0
network 172.16.0.0 ! ip default-gateway 172.16.31.1 ip
classless ip http server ! access-list 101 permit ip
19.19.19.0 0.0.0.255 18.18.18.0 0.0.0.255
!--- Access list that identifies the traffic to be
encrypted. In this case, !--- it is setting traffic from
the local Ethernet network to the remote !--- Ethernet
network. snmp-server manager ! line con 0 transport
input none line vty 0 4 password ww login ! end
```

Die Konfiguration des anderen Kabelmodems ist sehr ähnlich, sodass die meisten Kommentare in der vorherigen Konfiguration weggelassen werden.

uBR904-2

```
version 12.1
no service pad
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname ubr904-2
!
enable password ww
!
!
!
!
!
clock timezone - -8
ip subnet-zero
```

```
no ip finger
!
!
!
crypto isakmp policy 10
 hash md5
 authentication pre-share
 group 2
 lifetime 3600
crypto isakmp key mykey address 19.19.19.19
!
!
crypto IPSec transform-set TUNNELSET ah-md5-hmac ESP-Des
!
crypto map MYMAP local-address Ethernet0
crypto map MYMAP 10 ipsec-isakmp
 set peer 19.19.19.19
!--- Identifies the IP address for the destination peer
router. In this case, !--- the Ethernet interface of the
remote cable modem (uBR924-1) is used. set transform-set
TUNNELSET
 match address 101
!
!
!
!
interface Ethernet0
 ip address 18.18.18.18 255.255.255.0
 ip rip send version 2
 ip rip receive version 2
!
interface cable-modem0
 ip rip send version 2
 ip rip receive version 2
 no keepalive
 cable-modem downstream saved channel 555000000 42 1
 cable-modem Mac-timer t2 40000
 no cable-modem compliant bridge
 crypto map MYMAP
!
router rip
 version 2
 network 18.0.0.0
 network 172.16.0.0
!
ip default-gateway 172.16.30.1
ip classless
no ip http server
!
access-list 101 permit ip 18.18.18.0 0.0.0.255
19.19.19.0 0.0.0.255
snmp-server manager
!
line con 0
 transport input none
line vty 0 4
 password ww
 login
!
end
```

Auf dem CMTS uBR7246VXR wird auch Routing Information Protocol (RIP) Version 2 ausgeführt, sodass das Routing funktioniert. Dies ist die für das CMTS verwendete RIP-Konfiguration:

uBR7246VXR
router rip version 2 network 172.16.0.0 no auto-summary

Überprüfen

In diesem Abschnitt überprüfen Sie, ob Ihre Konfiguration ordnungsgemäß funktioniert.

So überprüfen Sie, ob IPsec funktioniert:

- Überprüfen Sie folgende Punkte:Die Cisco IOS-Software unterstützt IPsec.Die aktuelle Konfiguration ist korrekt.Schnittstellen sind aktiv.Routing funktioniert.Die zur Verschlüsselung des Datenverkehrs definierte Zugriffsliste ist korrekt.
- Erstellen Sie Datenverkehr, und sehen Sie sich die Encrypt und Decrypt an, um zu sehen, wie viel Prozent ansteigt.
- Aktivieren Sie die Debugging-Funktion für Krypto.

Das Output Interpreter Tool (nur registrierte Kunden) (OIT) unterstützt bestimmte **show**-Befehle. Verwenden Sie das OIT, um eine Analyse der **Ausgabe des** Befehls **show anzuzeigen**.

Führen Sie den Befehl **show version** auf beiden Kabelmodems aus.

```
ubr924-1#show version
Cisco Internetwork Operating System Software
IOS (tm) 920 Software (UBR920-K1O3SV4Y556I-M), Version 12.1(6),
RELEASE SOFTWARE (fc1)
Copyright (c) 1986-2000 by Cisco Systems, Inc.
Compiled Wed 27-Dec-00 16:36 by kellythw
Image text-base: 0x800100A0, data-base: 0x806C1C20

ROM: System Bootstrap, Version 12.0(6r)T3, RELEASE SOFTWARE (fc1)

ubr924-1 uptime is 1 hour, 47 minutes
System returned to ROM by reload at 10:39:05 - Fri Feb 9 2001
System restarted at 10:40:05 - Fri Feb 9 2001
System image file is "flash:ubr920-k1o3sv4y556i-mz.121-6"

cisco uBR920 CM (MPC850) processor (revision 3.e)
with 15872K/1024K bytes of memory.
Processor board ID FAA0422Q04F
Bridging software.
1 Ethernet/IEEE 802.3 interface(s)
1 Cable Modem network interface(s)
3968K bytes of processor board System flash (Read/Write)
1536K bytes of processor board Boot flash (Read/Write)

Configuration register is 0x2102
```

Auf dem uBR924-1 wird die Cisco IOS Software Release 12.1(6) mit dem VALUE SMALL OFFICE/VOICE/FW IPSec 56 Feature-Set ausgeführt.

```
ubr904-2#show version
Cisco Internetwork Operating System Software
```

```
IOS (TM) 900 Software (UBR900-K1OY556I-M), Version 12.1(6),
RELEASE SOFTWARE (fc1)
Copyright (c) 1986-2000 by cisco Systems, Inc.
Compiled Wed 27-DEC-00 11:06 by kellythw
Image text-base: 0x08004000, database: 0x085714DC

ROM: System Bootstrap, Version 11.2(19980518:195057), RELEASED SOFTWARE
ROM: 900 Software (UBR900-RBOOT-M), Version 11.3(11)NA,
EARLY DEPLOYMENT RELEASE SOFTWARE (fc1)

ubr904-2 uptime is 1 hour, 48 minutes
System returned to ROM by reload at 10:38:44 - Fri Feb 9 2001
System restarted at 10:40:37 - Fri Feb 9 2001
System image file is "flash:ubr900-k1oy556i-mz.121-6"

cisco uBR900 CM (68360) processor (revision D)
with 8192K bytes of memory.
Processor board ID FAA0235Q0ZS
Bridging software.
1 Ethernet/IEEE 802.3 interface(s)
1 Cable Modem network interface(s)
4096K bytes of processor board System flash (Read/Write)
2048K bytes of processor board Boot flash (Read/Write)

Configuration register is 0x2102
```

Auf dem uBR904-2 wird die Cisco IOS Software Release 12.1(6) mit dem Funktionssatz SMALL OFFICE/FW IPSec 56 ausgeführt.

```
ubr924-1#show ip interface brief
Interface              IP-Address      OK? Method Status          Protocol
Ethernet0              19.19.19.19     YES NVRAM  up              up
cable-modem0           172.16.31.20    YES unset  up              up

ubr904-2#show ip interface brief
Interface              IP-Address      OK? Method Status          Protocol
Ethernet0              18.18.18.18     YES NVRAM  up              up
cable-modem0           172.16.30.18    YES unset  up              up
```

Der letzte Befehl zeigt an, dass die Ethernet-Schnittstellen aktiv sind. Die IP-Adressen der Ethernet-Schnittstellen wurden manuell eingegeben. Die Kabelschnittstellen sind ebenfalls aktiv, und sie haben ihre IP-Adressen über DHCP gelernt. Da diese Kabeladressen dynamisch zugewiesen werden, können sie nicht als Peers in der IPSec-Konfiguration verwendet werden.

```
ubr924-1#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - ISIS level-1, L2 - ISIS level-2, ia - ISIS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is 172.16.31.1 to network 0.0.0.0

     19.0.0.0/24 is subnetted, 1 subnets
C       19.19.19.0 is directly connected, Ethernet0
R    18.0.0.0/8 [120/2] via 172.16.31.1, 00:00:23, cable-modem0
     172.16.0.0/16 is variably subnetted, 4 subnets, 3 masks
R       172.16.135.0/25 [120/1] via 172.16.31.1, 00:00:23, cable-modem0
```

```
R       172.16.29.0/27 [120/1] via 172.16.31.1, 00:00:23, cable-modem0
R       172.16.30.0/24 [120/1] via 172.16.31.1, 00:00:23, cable-modem0
C       172.16.31.0/24 is directly connected, cable-modem0
R    192.168.99.0/24 [120/3] via 172.16.31.1, 00:00:24, cable-modem0
     10.0.0.0/24 is subnetted, 2 subnets
R       10.10.10.0 [120/2] via 172.16.31.1, 00:00:24, cable-modem0
S*   0.0.0.0/0 [1/0] via 172.16.31.1
```

Aus dieser Ausgabe können Sie sehen, dass uBR924-1 die Route 18.18.18.0, die Ethernet-Schnittstelle von uBR904-2, erfährt.

```
ubr904-2#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - ISIS, L1 - ISIS level-1, L2 - ISIS level-2, IA - ISIS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is 172.16.30.1 to network 0.0.0.0

R    19.0.0.0/8 [120/2] via 172.16.30.1, 00:00:17, cable-modem0
     18.0.0.0/24 is subnetted, 1 subnets
C       18.18.18.0 is directly connected, Ethernet0
     172.16.0.0/16 is variably subnetted, 4 subnets, 3 masks
R       172.16.135.0/25 [120/1] via 172.16.30.1, 00:00:17, cable-modem0
R       172.16.29.224/27 [120/1] via 172.16.30.1, 00:00:17, cable-modem0
C       172.16.30.0/24 is directly connected, cable-modem0
R       172.16.31.0/24 [120/1] via 172.16.30.1, 00:00:17, cable-modem0
R    192.168.99.0/24 [120/3] via 172.16.30.1, 00:00:18, cable-modem0
     10.0.0.0/24 is subnetted, 1 subnets
R       10.10.10.0 [120/2] via 172.16.30.1, 00:00:18, cable-modem0
S*   0.0.0.0/0 [1/0] via 172.16.30.1
```

Aus der Routing-Tabelle von uBR904-2 können Sie sehen, dass das Netzwerk für das Ethernet von uBR924-1 in der Routing-Tabelle enthalten ist.

Hinweis: Es kann vorkommen, dass zwischen den beiden Kabelmodems kein Routing-Protokoll ausgeführt werden kann. In solchen Fällen müssen Sie dem CMTS statische Routen hinzufügen, um den Datenverkehr für die Ethernet-Schnittstellen der Kabelmodems zu leiten.

Als Nächstes muss die Zertifizierung der Zugriffsliste überprüft werden. auf beiden Routern den Befehl **show access-lists** ausführen.

```
ubr924-1#show access-lists
Extended IP access list 101
    permit ip 19.19.19.0 0.0.0.255 18.18.18.0 0.0.0.255 (2045 matches)

ubr904-2#show access-lists
Extended IP access list 101
    permit ip 18.18.18.0 0.0.0.255 19.19.19.0 0.0.0.255 (2059 matches)
```

Die Zugriffsliste legt die IPsec-Sitzung fest, wenn das LAN hinter uBR924-1 (19.19.19.0) IP-Datenverkehr an das LAN hinter uBR904-2 (18.18.18.0) sendet, und umgekehrt. Verwenden Sie in den Zugriffslisten *kein* "any", da dies Probleme verursacht. Weitere Informationen finden Sie unter Konfigurieren der IPsec-Netzwerksicherheit.

Es ist kein IPsec-Datenverkehr vorhanden. Geben Sie den Befehl **show crypto engine connection active** ein.

```
ubr924-1#show crypto engine connection active
ID Interface    IP-Address    State    Algorithm           Encrypt  Decrypt
1                              set     HMAC_MD5+DES_56_CB        0        0

ubr904-2#show crypto engine connection active
ID Interface    IP-Address    State    Algorithm           Encrypt  Decrypt
1                              set     HMAC_MD5+DES_56_CB        0        0
```

Es gibt keine IPsec-Verbindungen, da kein Datenverkehr mit den Zugriffslisten übereinstimmt.

Hinweis: Beachten Sie vor der Verwendung von **Debug-**Befehlen die Informationen zu Debug-Befehlen.

Im nächsten Schritt werden einige Krypto-Debugger aktiviert, um interessanten Datenverkehr zu generieren.

In diesem Beispiel sind diese Debugger aktiviert:

- **Debug-Krypto-Engine**
- **Debuggen von IPsec**
- **Schlüsselaustausch debug crypto**
- **debuggen crypto isakmp**

Sie müssen zunächst einen interessanten Datenverkehr generieren, um die Ausgabe der Debugger anzuzeigen. Stellen Sie einen erweiterten Ping-Befehl vom Ethernet-Port von uBR904-2 an den PC auf dem uBR924-1 aus (IP-Adresse 19.19.19.1).

```
ubr904-2#ping ip
Target IP address: 19.19.19.1
!--- IP address of PC1 behind the Ethernet of uBR924-1. Repeat count [5]: 100
!--- Sends 100 pings. Datagram size [100]: Timeout in seconds [2]: Extended commands [n]: y
Source address or interface: 18.18.18.18
!--- IP address of the Ethernet behind uBR904-2. Type of service [0]: Set DF bit in IP header?
[no]: Validate reply data? [no]: Data pattern [0xABCD]: Loose, Strict, Record, Timestamp,
Verbose[none]: Sweep range of sizes [n]: Type escape sequence to abort. Sending 100, 100-byte
ICMP Echos to 19.19.19.1, timeout is 2 seconds:
```

Im Beispiel uBR924-2 wird folgende Debugausgabe angezeigt:

```
ubr904-2#
01:50:37: IPSec(sa_request): ,
 (key eng. msg.) src= 18.18.18.18, dest= 19.19.19.19,
    src_proxy= 18.18.18.0/255.255.255.0/0/0 (type=4),
    dest_proxy= 19.19.19.0/255.255.255.0/0/0 (type=4),
    protocol= AH, transform= ah-md5-hmac ,
    lifedur= 3600s and 4608000kb,
    spi= 0x19911A16(428939798), conn_id= 0, keysize= 0, flags= 0x4004
01:50:37: IPSec(sa_request): ,
  (key Eng. msg.) src= 18.18.18.18, dest= 19.19.19.19,
    src_proxy= 18.18.18.0/255.255.255.0/0/0 (type=4),
    dest_proxy= 19.19.19.0/255.255.255.0/0/0 (type=4),
    protocol= ESP, transform= ESP-Des ,
    lifedur= 3600s and 4608000kb,
    spi= 0x7091981(118036865), conn_id= 0, keysize= 0, flags= 0x4004
01:50:37: ISAKMP: received ke message (1/2)
01:50:37: ISAKMP (0:1): sitting IDLE. Starting QM immediately (QM_IDLE)
01:50:37: ISAKMP (0:1): beginning Quick Mode exchange, M-ID of 1108017901
01:50:37: CryptoEngine0: generate hmac context for conn id 1
```

```
01:50:37: ISAKMP (1): sending packet to 19.19.19.19 (I) QM_IDLE
01:50:37: ISAKMP (1): received packet from 19.19.19.19 (I) QM_IDLE
01:50:37: CryptoEngine0: generate hmac context for conn id 1
01:50:37: ISAKMP (0:1): processing SA payload. message ID = 1108017901
01:50:37: ISAKMP (0:1): Checking IPSec proposal 1
01:50:37: ISAKMP: transform 1, AH_MD5
01:50:37:   attributes in transform:
01:50:3.!!!!!!!!!!!!!!!!!!!!!!!7: ISAKMP:      encaps is 1
01:50:37: ISAKMP:      SA life type in seconds
01:50:37: ISAKMP:      SA life duration (basic) of 3600
01:50:37: ISAKMP:      SA life type in kilobytes
01:50:37: ISAKMP:      SA life duration (VPI) of  0x0 0x46 0x50 0x0
01:50:37: ISAKMP:      authenticator is HMAC-MD5
01:50:37: validate proposal 0
01:50:37: ISAKMP (0:1): atts are acceptable.
01:50:37: ISAKMP (0:1): Checking IPSec proposal 1
01:50:37: ISAKMP: transform 1, ESP_DES
01:50:37:   attributes in transform:
01:50:37: ISAKMP:      encaps is 1
01:50:37: ISAKMP:      SA life type in seconds
01:50:37: ISAKMP:      SA life duration (basic) of 3600
01:50:37: ISAKMP:      SA life type in kilobytes
01:50:37: ISAKMP:      SA life duration (VPI) of  0x0 0x46 0x50 0x0
01:50:37: validate proposal 0
01:50:37: ISAKMP (0:1): atts are acceptable.
01:50:37: IPSec(validate_proposal_request): proposal part #1,
  (key Eng. msg.) dest= 19.19.19.19, src= 18.18.18.18,
  dest_proxy= 19.19.1!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Success rate is 99 percent (99/100), round-trip min/avg/max = 30/40/70 ms
ubr904-2#
```

Beachten Sie, dass der erste Ping fehlgeschlagen ist. Dies liegt daran, dass die Verbindung hergestellt werden muss.

Der uBR924-1 zeigt diese Debugausgabe:

```
ubr924-1#
01:50:24: ISAKMP (1): received packet from 18.18.18.18 (R) QM_IDLE
01:50:24: CryptoEngine0: generate hmac context for conn id 1
01:50:24: ISAKMP (0:1): processing SA payload. Message ID = 1108017901
01:50:24: ISAKMP (0:1): Checking IPSec proposal 1
01:50:24: ISAKMP: transform 1, AH_MD5
01:50:24:   attributes in transform:
01:50:24:      encaps is 1
01:50:24:      SA life type in seconds
01:50:24:      SA life duration (basic) of 3600
01:50:24:      SA life type in kilobytes
01:50:24:      SA life duration (VPI) of  0x0 0x46 0x50 0x0
01:50:24:      authenticator is HMAC-MD5
01:50:24: validate proposal 0
01:50:24: ISAKMP (0:1): atts are acceptable.
01:50:24: ISAKMP (0:1): Checking IPSec proposal 1
01:50:24: ISAKMP: transform 1, ESP_DES
01:50:24:   attributes in transform:
01:50:24:      encaps is 1
01:50:24:      SA life type in seconds
01:50:24:      SA life duration (basic) of 3600
01:50:24:      SA life type in kilobytes
01:50:24:      SA life duration (VPI) of  0x0 0x46 0x50 0x0
01:50:24: validate proposal 0
01:50:24: ISAKMP (0:1): atts are acceptable.
```

```
01:50:24: IPSec(validate_proposal_request): proposal part #1,
  (key Eng. msg.) dest= 19.19.19.19, src= 18.18.18.18,
    dest_proxy= 19.19.19.0/255.255.255.0/0/0 (type=4),
    src_proxy= 18.18.18.0/255.255.255.0/0/0 (type=4),
    protocol= AH, transform= ah-md5-hmac ,
    lifedur= 0s and 0kb,
    spi= 0x0(0), conn_id= 0, keysize= 0, flags= 0x4
01:50:24: IPSec(validate_proposal_request): proposal part #2,
  (key Eng. msg.) dest= 19.19.19.19, src= 18.18.18.18,
    dest_proxy= 19.19.19.0/255.255.255.0/0/0 (type=4),
    src_proxy= 18.18.18.0/255.255.255.0/0/0 (type=4),
    protocol= ESP, transform= ESP-Des ,
    lifedur= 0s and 0kb,
    spi= 0x0(0), conn_id= 0, keysize= 0, flags= 0x4
01:50:24: validate proposal request 0
01:50:24: ISAKMP (0:1): processing NONCE payload. Message ID = 1108017901
01:50:24: ISAKMP (0:1): processing ID payload. Message ID = 1108017901
01:50:24: ISAKMP (1): ID_IPV4_ADDR_SUBNET src 18.18.18.0/255.255.255.0
   prot 0 Port 0
01:50:24: ISAKMP (0:1): processing ID payload. Message ID = 1108017901
01:50:24: ISAKMP (1): ID_IPV4_ADDR_SUBNET dst 19.19.19.0/255.255.255.0
   prot 0 Port 0
01:50:24: ISAKMP (0:1): asking for 2 spis from IPSec
01:50:24: IPSec(key_engine): got a queue event...
01:50:24: IPSec(spi_response): getting spi 393021796 for SA
        from 18.18.18.18     to 19.19.19.19     for prot 2
01:50:24: IPSec(spi_response): getting spi 45686884 for SA
        from 18.18.18.18     to 19.19.19.19     for prot 3
01:50:24: ISAKMP: received ke message (2/2)
01:50:24: CryptoEngine0: generate hmac context for conn id 1
01:50:24: ISAKMP (1): sending packet to 18.18.18.18 (R) QM_IDLE
01:50:24: ISAKMP (1): received packet from 18.18.18.18 (R) QM_IDLE
01:50:24: CryptoEngine0: generate hmac context for conn id 1
01:50:24: IPSec allocate flow 0
01:50:24: IPSec allocate flow 0
01:50:24: ISAKMP (0:1): Creating IPSec SAs
01:50:24:         inbound SA from 18.18.18.18 to 19.19.19.19
                  (proxy 18.18.18.0 to 19.19.19.0)
01:50:24:         has spi 393021796 and conn_id 2000 and flags 4
01:50:24:         lifetime of 3600 seconds
01:50:24:         lifetime of 4608000 kilobytes
01:50:24:         outbound SA from 19.19.19.19 to 18.18.18.18
                  (proxy 19.19.19.0 to 18.18.18.0)
01:50:24:         has spi 428939798 and conn_id 2001 and flags 4
01:50:24:         lifetime of 3600 seconds
01:50:24:         lifetime of 4608000 kilobytes
01:50:24: ISAKMP (0:1): Creating IPSec SAs
01:50:24:         inbound SA from 18.18.18.18 to 19.19.19.19
                  (proxy 18.18.18.0 to 19.19.19.0)
01:50:24:         has spi 45686884 and conn_id 2002 and flags 4
01:50:24:         lifetime of 3600 seconds
01:50:24:         lifetime of 4608000 kilobytes
01:50:24:         outbound SA from 19.19.19.19 to 18.18.18.18
                  (proxy 19.19.19.0 to 18.18.18.0)
01:50:24:         has spi 118036865 and conn_id 2003 and flags 4
01:50:25:         lifetime of 3600 seconds
01:50:25:         lifetime of 4608000 kilobytes
01:50:25: ISAKMP (0:1): deleting node 1108017901 error FALSE reason
          "quick mode done (await()"
01:50:25: IPSec(key_engine): got a queue event...
01:50:25: IPSec(initialize_sas): ,
  (key Eng. msg.) dest= 19.19.19.19, src= 18.18.18.18,
    dest_proxy= 19.19.19.0/255.255.255.0/0/0 (type=4),
    src_proxy= 18.18.18.0/255.255.255.0/0/0 (type=4),
```

```
    protocol= AH, transform= ah-md5-hmac ,
    lifedur= 3600s and 4608000kb,
    spi= 0x176D0964(393021796), conn_id= 2000, keysize= 0, flags= 0x4
01:50:25: IPSec(initialize_sas): ,
  (key Eng. msg.) src= 19.19.19.19, dest= 18.18.18.18,
    src_proxy= 19.19.19.0/255.255.255.0/0/0 (type=4),
    dest_proxy= 18.18.18.0/255.255.255.0/0/0 (type=4),
    protocol= AH, transform= ah-md5-hmac ,
    lifedur= 3600s and 4608000kb,
    spi= 0x19911A16(428939798), conn_id= 2001, keysize= 0, flags= 0x4
01:50:25: IPSec(initialize_sas): ,
  (key Eng. msg.) dest= 19.19.19.19, src= 18.18.18.18,
    dest_proxy= 19.19.19.0/255.255.255.0/0/0 (type=4),
    src_proxy= 18.18.18.0/255.255.255.0/0/0 (type=4),
    protocol= ESP, transform= ESP-Des ,
    lifedur= 3600s and 4608000kb,
    spi= 0x2B92064(45686884), conn_id= 2002, keysize= 0, flags= 0x4
01:50:25: IPSec(initialize_sas): ,
  (key Eng. msg.) src= 19.19.19.19, dest= 18.18.18.18,
    src_proxy= 19.19.19.0/255.255.255.0/0/0 (type=4),
    dest_proxy= 18.18.18.0/255.255.255.0/0/0 (type=4),
    protocol= ESP, transform= ESP-Des ,
    lifedur= 3600s and 4608000kb,
    spi= 0x7091981(118036865), conn_id= 2003, keysize= 0, flags= 0x4
01:50:25: IPSec(create_sa): sa created,
  (sa) sa_dest= 19.19.19.19, sa_prot= 51,
    sa_spi= 0x176D0964(393021796),
    sa_trans= ah-md5-hmac , sa_conn_id= 2000
01:50:25: IPSec(create_sa): sa created,
  (sa) sa_dest= 18.18.18.18, sa_prot= 51,
    sa_spi= 0x19911A16(428939798),
    sa_trans= ah-md5-hmac , sa_conn_id= 2001
01:50:25: IPSec(create_sa): sa created,
  (sa) sa_dest= 19.19.19.19, sa_prot= 50,
    sa_spi= 0x2B92064(45686884),
    sa_trans= ESP-Des , sa_conn_id= 2002
01:50:25: IPSec(create_sa): sa created,
  (sa) sa_dest= 18.18.18.18, sa_prot= 50,
    sa_spi= 0x7091981(118036865),
    sa_trans= ESP-Des , sa_conn_id= 2003
ubr924-1#
```

Sobald der IPsec-Tunnel erstellt wurde, können Sie die Verbindung und die verschlüsselten und entschlüsselten Pakete sehen.

```
ubr924-1#show crypto engine connection active
ID   Interface      IP-Address     State  Algorithm          Encrypt  Decrypt
   1                                set    HMAC_MD5+DES_56_CB       0        0
2000 cable-modem0   172.16.31.20   set    HMAC_MD5                 0       99
2001 cable-modem0   172.16.31.20   set    HMAC_MD5                99        0
2002 cable-modem0   172.16.31.20   set    DES_56_CBC               0       99
2003 cable-modem0   172.16.31.20   set    DES_56_CBC              99        0
```

Die erste 200-fache Leitung zeigt die 99 empfangenen Pakete an. Sie muss die Pakete entschlüsseln, um sie an PC1 zu senden. Die zweite Zeile zeigt 99 gesendete Pakete. Sie muss die Pakete verschlüsseln, bevor sie an uBR904-2 gesendet werden. Die dritte und vierte Zeile führen den gleichen Prozess aus, jedoch mit ESP-DES-Transformation anstelle von AH-MD5-HMAC.

Hinweis: Wenn der auf dem Kabelmodem konfigurierte Transformationssatz ESP-DES ESP-MD5-HMAC ist, werden nur zwei autonome Systeme (ASs) angezeigt, im Gegensatz zu den vier, die im

vorherigen **Befehl show** gezeigt werden.

```
ubr904-2#show crypto engine connection active
ID   Interface      IP-Address     State  Algorithm          Encrypt  Decrypt
   1                                set    HMAC_MD5+DES_56_CB       0        0
2000 cable-modem0   172.16.30.18   set    HMAC_MD5                 0       99
2001 cable-modem0   172.16.30.18   set    HMAC_MD5                99        0
2002 cable-modem0   172.16.30.18   set    DES_56_CBC               0       99
2003 cable-modem0   172.16.30.18   set    DES_56_CBC              99        0
```

Stellen Sie einen erweiterten Ping-Befehl vom uBR924-1 an PC2 aus, um festzustellen, ob die Zähler für die verschlüsselten und entschlüsselten Pakete inkrementell erhöhen.

```
ubr924-1#ping ip
Target IP address: 18.18.18.1
Repeat count [5]: 50
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]: y
Source address or interface: 19.19.19.19
Type of service [0]:
Set DF bit in IP header? [no]:
Validate reply data? [no]:
Data pattern [0xABCD]:
Loose, Strict, Record, Timestamp, Verbose[none]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 50, 100-byte ICMP Echos to 18.18.18.1, timeout is 2 seconds:
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Success rate is 100 percent (50/50), round-trip min/avg/max = 28/30/33 ms

ubr924-1#show crypto engine connection active
ID   Interface      IP-Address     State  Algorithm          Encrypt  Decrypt
   1                                set    HMAC_MD5+DES_56_CB       0        0
2000 cable-modem0   172.16.31.20   set    HMAC_MD5                 0      149
2001 cable-modem0   172.16.31.20   set    HMAC_MD5               149        0
2002 cable-modem0   172.16.31.20   set    DES_56_CBC               0      149
2003 cable-modem0   172.16.31.20   set    DES_56_CBC             149        0

ubr904-2#show crypto engine connection active
ID   Interface      IP-Address     State  Algorithm          Encrypt  Decrypt
   1                                set    HMAC_MD5+DES_56_CB       0        0
2000 cable-modem0   172.16.30.18   set    HMAC_MD5                 0      149
2001 cable-modem0   172.16.30.18   set    HMAC_MD5               149        0
2002 cable-modem0   172.16.30.18   set    DES_56_CBC               0      149
2003 cable-modem0   172.16.30.18   set    DES_56_CBC             149        0
```

Ein weiterer erweiterter Ping-Befehl kann ausgegeben werden, um zu sehen, dass die Zähler wieder inkrementiert werden. Diesmal senden Sie einen Ping-Befehl mit 500 Paketen von uBR904-2 an die Ethernet-Schnittstelle uBR924-1 (19.19.19.19).

```
ubr904-2#ping ip
Target IP address: 19.19.19.19
Repeat count [5]: 500
Datagram size [100]: 1000
Timeout in seconds [2]:
Extended commands [n]: y
```

```
Source address or interface: 18.18.18.18
Type of service [0]:
Set DF bit in IP header? [no]:
Validate reply data? [no]:
Data pattern [0xABCD]:
Loose, Strict, Record, Timestamp, Verbose[none]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 500, 1000-byte ICMP Echos to 19.19.19.19, timeout is 2 seconds:
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
01:59:06: IPSec(encapsulate): encaps area too small, moving to new buffer:
idbtype 0, encaps_size 26, header size 60, avail 84!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!
Success rate is 100 percent (500/500), round-trip min/avg/max = 98/135/352 ms

ubr904-2#show crypto engine connection active
ID   Interface       IP-Address     State  Algorithm          Encrypt  Decrypt
   1                                set    HMAC_MD5+DES_56_CB       0        0
2000 cable-modem0    172.16.30.18   set    HMAC_MD5                 0      649
2001 cable-modem0    172.16.30.18   set    HMAC_MD5               649        0
2002 cable-modem0    172.16.30.18   set    DES_56_CBC               0      649
2003 cable-modem0    172.16.30.18   set    DES_56_CBC             649        0

ubr924-1#show crypto engine connection active
ID   Interface       IP-Address     State  Algorithm          Encrypt  Decrypt
   1                                set    HMAC_MD5+DES_56_CB       0        0
2000 cable-modem0    172.16.31.20   set    HMAC_MD5                 0      649
2001 cable-modem0    172.16.31.20   set    HMAC_MD5               649        0
2002 cable-modem0    172.16.31.20   set    DES_56_CBC               0      649
2003 cable-modem0    172.16.31.20   set    DES_56_CBC             649        0
```

Sie können die Befehle **clear crypto isakmp** und **clear crypto sa** zum Löschen der Verbindungen ausgeben. Wenn während der Ablaufzeit kein Datenverkehr über den IPsec-Tunnel läuft, setzt IPsec die Verbindung automatisch zurück.

Fehlerbehebung

Zur Behebung dieser Konfiguration sind derzeit keine spezifischen Informationen verfügbar.

Zugehörige Informationen

- IPsec-Netzwerksicherheitsbefehle
- Einführung in die IP-Sicherheit (IPsec)-Verschlüsselung - Debuginformationen
- IPsec-Konfigurationsbeispiele
- Konfigurieren der IPsec-Netzwerksicherheit
- Konfigurieren der Cisco Cable Access Router der Serie uBR900
- Cisco Kabel-/Breitbanddownloads (nur registrierte Kunden)
- Unterstützung von Breitbandkabeltechnologie
- Technischer Support und Dokumentation - Cisco Systems